前　言

近年来，模糊逻辑的应用迅速发展，其应用范围从照相机、便携式摄像机、洗衣机和微波炉等日用消费品，直到工业控制、医疗器械、决策支持系统和人力资源管理等领域。尤其在家用电器领域，无论对专家还是对普通消费者，模糊逻辑都给人以深刻的印象。我国已经加入世界贸易组织，各种工业产品尤其是家电产品越来越向着智能化、人性化的方向发展。从在校大学生到从事科研、生产的工程技术人员，都对模糊逻辑产生了浓厚的兴趣。但由于对模糊逻辑的学习或研究具有一定的困难，而且难于方便地对模糊推理系统进行实验和验证。因此，许多人都对模糊逻辑既有兴趣又望而却步。

本书的出版可以解决这一问题。本书在第1章中对模糊推理与模糊控制系统作了深入浅出的介绍，既可使从未接触过模糊逻辑的读者了解模糊逻辑，也可使学习过模糊逻辑的读者对模糊逻辑与模糊推理以及模糊控制系统有更深入的理解，并了解其现状及发展动态。在第2章中简要介绍了MATLAB模糊逻辑工具箱。MATLAB是一个功能强大的应用软件，模糊逻辑工具箱是其中的一个工具箱。在MATLAB上可采用两种方式，即用户交互式界面和命令行方式，进行模糊逻辑的学习与研究，并对各种从简单到复杂的模糊推理系统进行实验和验证。在第3章和第4章中结合实例，对基于MATLAB语言的模糊逻辑应用和模糊逻辑工具箱的使用作了详细的介绍，并给出了使用的详细步骤及大量的应用实例。本书的特点是力求避开大而全的数学理论和繁琐的数学推导，而尽量从工程应用与实践的角度来阐述模糊推理的学习及其在MATLAB模糊逻辑工具箱中的实现，使得读者可以尽快地掌握模糊逻辑的内容与模糊控制的实现和使用。本书既可作为大学高年级学生自学模糊控制和智能控制的自学教材，又可作为大学教师教授相关课程的参考书，尤其可作为工程技术人员学习和实现模糊控制与智能控制的工具书。全书内容结构清晰、简明易懂，便于查找。

在本书的写作过程中，得到了西安交通大学出版社的热情支持，同时沈永福、张翼飞、梁勇等计算了本书的部分例子，在此一并致谢。

由于模糊控制理论是一个新颖的领域，目前仍处于发展阶段，许多理论还不够完善，加之作者水平有限，在模糊控制方面的工程实践尚不够丰富，所以对书中不妥之处，恳请读者予以批评指正。

作者

2002年2月

目　　录



第1章 模糊控制系统

1.1 模糊逻辑的历史

L. A. Zadeh 于 20 世纪 60 年代引入模糊逻辑，以表示并利用模糊的和不确定的知识。在他的《不相容原理》一书中，有如下的论述：

随着系统的复杂程度不断提高，人们对其精确而有意义地描述的能力不断地降低，以至在达到某一个阈值之后，系统的精确性和复杂性之间呈现出几乎是相互完全排斥的性质。

因此他认为，应当引入一种新的方法，使新开发的日趋复杂的系统能够具有"人情味"。1965 年，他首次提出了"模糊集合"的概念；1973 年，他又进一步研究了模糊语言处理，提出了用模糊语言进行系统描述的方法，给出了模糊推理的理论基础，并为模糊控制的实施提供了有效的手段。1974 年，A. Mamdani 和他的学生在 Queen Mary 学院首次应用模糊逻辑，实现了蒸汽发动机的模糊控制实验。自此，模糊逻辑在控制工程中的应用有了一个飞速发展的历史。

模糊逻辑是对模糊的、自然语言的表达和描述进行操作与利用。它允许在模糊系统中纳入常识和自学习规则，并意味着一个学习模块能够用一个模糊规则集合来解释其行为。因此模糊系统对使用者来说是透明的（transparent），这与许多人工神经网络（ANNs，Artificial Neural Networks）形成直接的对比。

同许多人工智能（AI，Artificial Intelligence）算法一样（模糊逻辑也可被视为一种人工智能技术），模糊逻辑最初应用到那些由人来执行显然毫不费力，但对基于传统算法的方法而言却很困难的课题上，例如语音识别、部分模式匹配、视觉数据分析等等。模糊逻辑的发展是由于人们需要一种对不完全、不精确信息做出决定的方法而被激发的，但模糊产品应用得最多的领域则是工程系统。这大概是由于对许多的模糊建模和控制系统，只需要使用少量的、易于理解的模糊信息处理技术就可以完成，因此获得这些算法的学习和归纳能力的内在信息是有可能的。

20 世纪 70 年代，大量的静态模糊控制器被开发出来，并生产出第一个能够改变其规则以提高性能的自组织模糊控制器。1980 年开始了模糊控制在应用领域的研究。20 世纪 80 年代初，日本和东亚许多主要的电子和自动化公司开始对模糊控制感兴趣，在此后及 90 年代早期，大量的基于模糊控制的消费产品问世，例如洗衣机、空调器、驾驶操作系统等。这些家电产品在节约资源、方便使用以及使用效果方面更富有"人情味"，也更符合人的实际生活。与此同时，各种各样的工业模糊控制系统也被研制成功。如各种熔炉、电气炉以及水泥生成炉的控制系统、核能发电供水系统、金属板成形控制系统、汽车的控制系统、机器人控制系统，以及航空、通讯领域的专家系统等。这些系统中的大部分是静态的，而且这些模糊系统的成功主要源于模糊逻辑在表示和操作上的模糊性。在这当中，专家知识被用以生成复杂的非线性控制界面。在被预置了一组庞大的模糊规则之后，用一个反复迭代的程序开发这类系统。其中重要的一点是，当前开发和应用的模糊系统主要是静态的。模糊逻辑最重要的性质之一，即是静态模糊系统同启发式规则相结合，从而具备了解决不确定性问题的能力。

模糊理论于 20 世纪 70 年代后期才引入我国。1981 年，我国创办了当时世界上第二个专门刊物《模糊数学》，即后来的《模糊系统与数学》。1982 年，我国成立了全国模糊系统与数学学会。目前许多高等院校已开设了模糊数学课程，相继建立了硕士点、博士点。Zadeh 不止一次地评价中国的模糊理论与应用研究队伍属于国际四支劲旅之一。

对许多难于建模和控制的系统，模糊逻辑以及应用模糊逻辑的模糊系统是非常有用的。本书将力求简单地讨论这些问题，并说明应用这些技术可能存在的问题以及克服这些困难的解决方案。

1.2　模糊集

要理解模糊逻辑与模糊控制系统，首先需要了解模糊集的概念。

模糊集的概念与古典集的概念相对应。模糊集既区别于古典集，又与古典集有密切的联系。古典集用于描述"非此即彼"的清晰概念。对于一个古典集合，一个给定的元素要么属于它，要么不属于它。而模糊集用于描述一个没有明确、清楚地定义界线的集合，即它包含的元素可以部分地隶属于这个集合。

例如，"所有高于 1.6 米的人"，这是一个清晰的概念。它表明凡是高于 1.6 米的人都是该集合的成员。尽管其元素无法一一列举，但其范围是可完全确定的。但是，若将上述概念改为"所有比 1.6 米高得多的人"，这就变成一个模糊概念了。因为无法划出严格分明的界限，使得在此界限内的人都属于"比 1.6 米高得多的人"，否则都不属于。而只能说某个人属于"比 1.6 米高得多的人"这个集合的程度高，另一个人属于它的程度低。因此，对于模糊概念而言，不能仿照清晰概念用"属于或不属于"来表述。

模糊概念来源于自然界中客观存在的模糊现象。人们在了解、掌握和处理自然现象时，大脑中所形成的概念往往是模糊的。这些概念的类属边界是不清晰的。由此产生的划分、判断与推理也都具有模糊性。人类的大脑具有很高的模糊划分、模糊判断和模糊推理的能力。人们的自然语言是为了表达和传递知识，在其中已巧妙地渗透着模糊性，并用尽可能少的词汇表达尽可能多的意思。事实上，人们大多数的推理和概念的形成是与使用模糊逻辑和模糊规则联系在一起的。在这个意义上，模糊逻辑既是旧的又是新的，因为，尽管模糊逻辑作为一门现代系统科学的时间还很短，但模糊逻辑的概念其实早就存在于我们的生活中了。

Zadeh 指出："在模糊逻辑语言变量的基本概念中，其变量的值是一个'词'而非'数'。实际上，模糊逻辑的大部分内容都可以作为一种用'词'而非'数'来进行计算的方法学来看待。虽然在内涵上，'词'不如'数'精确，但它更接近人的直觉。而且，用'词'来计算放宽了对不精确量的容许限，从而降低了解算所需的花费。"为了表示例如"小"这样的"词"(在后面我们按使用惯例统称为"语言值")，Zadeh 提出了模糊集的概念。一个典型的古典集是与一个特征函数(characteristic function)相联系的。当一个元素属于该集合时，特征函数的值为 1，否则特征函数的值为 0。因此，要描述一个集合，可以明确地写出其元素，也可以由定义其特征函数来说明。Zadeh 扩展了这种二值特征函数的思想，提出了隶属函数的概念，使元素可以部分地属于一个集合，而隶属函数(membership function)返回一个处于单位区间[0,1]的值。因此输入可以是部分地属于一个集合的元素。

更正式的说法是，模糊集 A 是定义在一个在输入 ξ 之上并由其隶属函数 $\mu_A(\cdot):\xi\to[0,1]$ 表征的集合。输入域可以是离散的也可是连续的，但对许多的建模与控制应用问题而言，连续表示更合适一些。

一般地，可以用 3 种方法来表示一个模糊集。

方法 1　Zadeh 记法

假设 ξ 是一个普通集合，称为论域。从 ξ 到区间[0,1]的映射 A 称为 ξ 上的一个模糊集合。μ_A 表示 ξ 隶属于模糊集合 A 的程度，称为隶属度。$\mu_A(\cdot)$ 称为隶属函数。若 ξ 为离散集合，则可表示为

$$\mu_A = \sum(\mu_A(\cdot)/\xi)$$

若 ξ 不是离散集合，则可表示为

$$\mu_A = \int(\mu_A(\cdot)/\xi)$$

在此的积分与累加符号与其通常意义不同，它表示的是各个元素与其隶属度对应关系的一个总括。

方法 2　序偶集合记法

将 μ_A 写成序偶的集合

$$\mu_A = \{(\xi_1, \mu_A(\xi_1)), (\xi_2, \mu_A(\xi_2)), \cdots, (\xi_n, \mu_A(\xi_n))\}$$

其中的每一元素是个序偶$(\xi_i, \mu_A(\xi_i))$。第一分量表示论域中的元素,第二分量为相应元素的隶属度。

方法 3 模糊向量记法

将 μ_A 写成向量的形式,称之为模糊向量

$$\mu_A = (\mu_A(\xi_1), \mu_A(\xi_2), \cdots, \mu_A(\xi_n))$$

要理解模糊逻辑的基本含义,需要记住的是,在模糊逻辑中,任何的表述在实质上都是一个度量。

下面我们来看两个例子。

例 1.1 关于一周中的每一天属于"周末"的度量。

考虑用图 1-1 对"周末"进行分级。

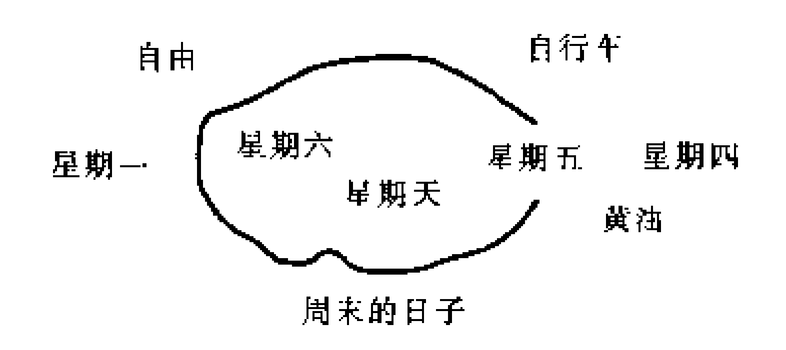


图 1-1 "周末"的集合

任何人都会将星期六和星期天划到"周末"的范畴,但对星期五则难以划分。它似乎应当属于周末的,但在某种程度上,从技术上又似乎应当将它从周末中排除。因此在图中星期五处于"骑墙"的位置。

在这种情况下,简单边界的二值(YES-NO)逻辑不再适用。生活中,人们往往用一个不太肯定的数字来做判断。例如

问:星期六是周末吗?

答:1(是,或真)

问:星期二是周末吗?

答:0(不,或假)

问:星期五是周末吗?

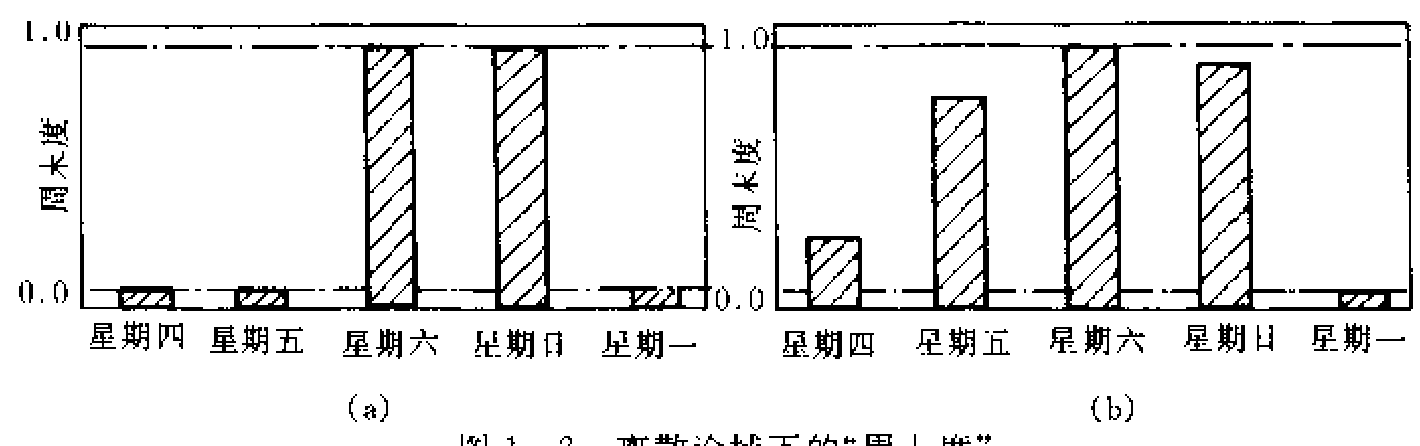
答:0.8(差不多是,但不完全是)

问:星期天是周末吗?

答:0.95(是,但和星期六还不完全一样)

如果我们必须绝对地说"是"或"非",则在图 1-2(a)的两值逻辑的周末图中表现了关于

“周末度量”的真实值；如果我们可以用模糊的中间值来回答，则在图 1－2(b)的多值逻辑的周末图中表现了关于“周末度量”的真实值。

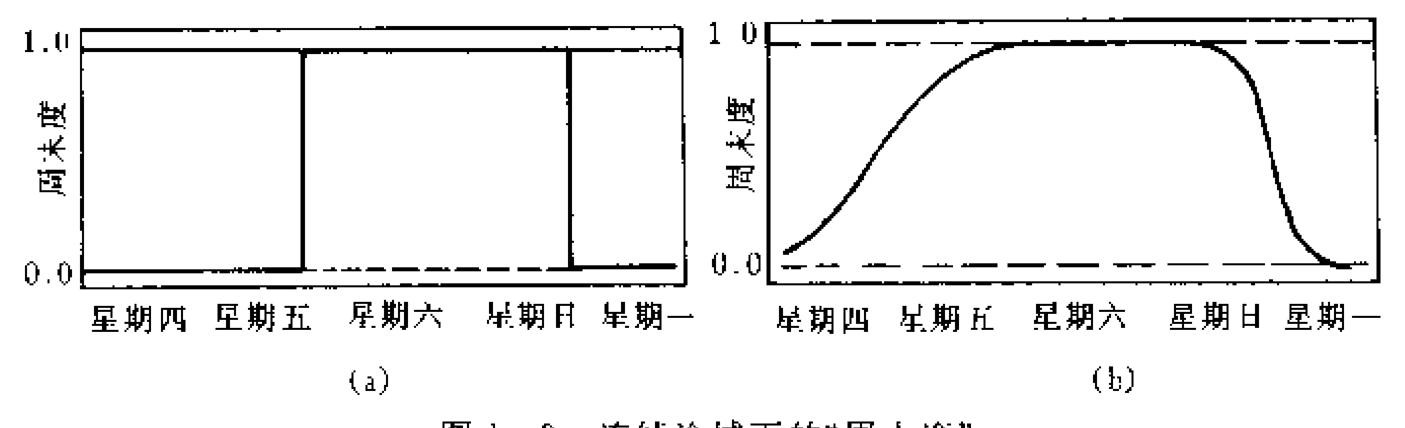


(a)　(b)

图 1－2　离散论域下的“周末度”

(a) 两值逻辑的“周末”；(b) 多值逻辑的“周末”

在上图中，我们实质上给出的是输入域为离散时，各天属于周末的度量。在图 1－3 中我们给出输入域为连续时的周末度量。



(a)　(b)

图 1－3　连续论域下的“周末度”

(a) 两值逻辑的“周末”；(b) 多值逻辑的“周末”

由于图形是连续的，我们现在定义了每一时刻(而不是一整天)隶属于周末的度量。在两值逻辑周末度的图中，注意在星期五的半夜，只要一过 12 点，周末值就立即非连续地从 0 跳转到 1。从统计学的观点而言，这种定义周末的方法是非常有用的，但它与我们在真实世界中的周末概念却没有确切的联系。在多值逻辑周末度的图中，我们看到的是一条平滑变化的曲线，它当中含有星期五的一整天，在某个较小的程度上，星期五的一部分属于星期四，另一部分属于周末。因此它应当是周末模糊集的部分成员。

例 1.2　关于一年中的每一天属于某个“季节”的度量。

在北半球，夏季正式开始的准确时刻是在每年的 6 月底。用天文学来定义季节，我们得到如图 1－4(a)有明显边界的图形。但是，我们对季节变化的经验却是如图 1－4(b)中所示的连续图形(按北半球气候的温度划分)。

现在，我们已经可以更清楚地理解到前面所讨论过的，即模糊集是一个没有明确、清楚的

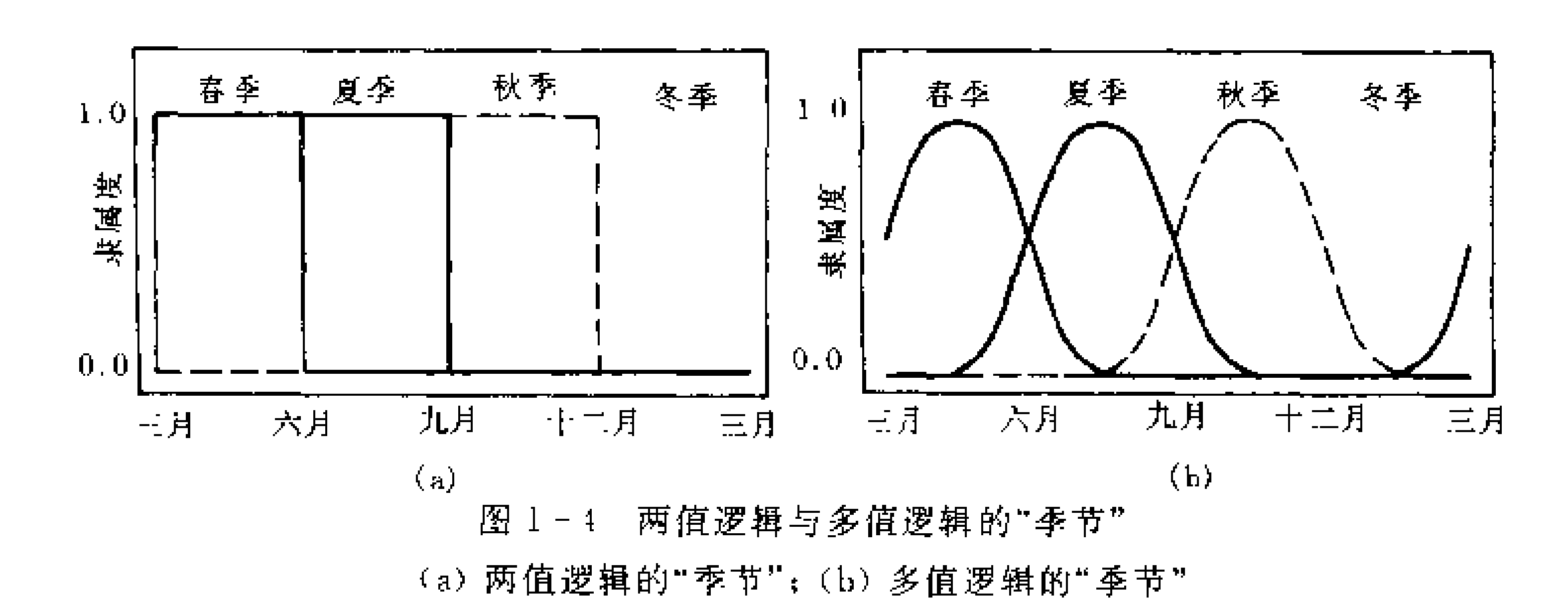


图 1-4 两值逻辑与多值逻辑的"季节"
(a) 两值逻辑的"季节"; (b) 多值逻辑的"季节"

定义界线的集合。对一个模糊集而言,它包含的元素可以部分地隶属于这个集合。例如,正如前面所说,在模糊逻辑中,任何表述在实质上都变成一个度量。星期四属于周末的度量大约是0.3,星期五属于周末的度量则大约是0.7;而六月份属于春季的度量大约是0.5,属于夏季的度量大约也是0.5,而它属于秋季的度量则为0。

在前面所说的定义一周中任一时刻的周末度的曲线,是一个映射输入空间(一周中的时间)到输出空间(周末)的函数;而定义一年中的任一天属于某一个季节的曲线,同样是一个映射输入空间(一年中的任一天)到输出空间(季节)的函数。具体地说,曲线上的任一点都表示一个隶属度,而描述这条曲线的函数则称为隶属函数。模糊集是定义在一个输入空间之上并由其隶属函数表征的集合。

"模糊集"与"隶属函数"之间的关系非常紧密。在许多场合,"模糊集"与"隶属函数"两个词的意义是统一的,甚至可以互换使用。但在应用中,关于隶属函数的意义及其确定仍须进行大量的具体工作。因此下面我们进行关于隶属函数的详细说明。

1.3 隶属函数

隶属函数(MF,Membership Fuction)是一条曲线,它定义了怎样将输入空间(又称"论域")上的每一点映射到一个0到1之间的隶属度。

比如,定义周末度的曲线和定义季节的曲线,都可用隶属函数进行描述。输出轴上的值在0到1之间,称为隶属度。输出曲线则称为隶属函数曲线(或简称隶属函数)。

隶属函数常由指定的μ给出。隶属函数惟一的必须满足的条件,是它的值必须在0到1之间变化。函数本身可以是一条任意的曲线,曲线的形状应当满足在某种角度下,我们可以简单、方便、快速、高效地将其定义为一个函数。

1.3.1　隶属函数的几种确定方法

从表面上看，隶属度似乎是主观的。实际上，模糊性的根源在于客观事物差异之间存在着中间过渡，存在着亦此亦彼的现象。但是，在亦此亦彼中依然存在着差异，依然可以相互比较，在上一层次中是亦此亦彼的东西，在下一层次中又可能是非此即彼的东西。这样，在客观上对隶属度进行了某种限定，使得不能主观任意地捏造隶属度。因此，隶属函数也是具有客观规律的东西，不能由主观任意确定。一般地，确定隶属函数主要有以下几种方法。

1. 模糊统计法

在某些场合下，隶属度可用模糊统计的方法来确定。模糊统计试验的目的，是用确定性的手段去研究隶属度的不确定性。

模糊统计的特点，是在每次试验中，元素ξ是固定的，而集合A'是可变的。记A'为论域中一个可变的普通集合。A'按照某种条件与一个模糊集合A相联系。模糊统计试验的基本要求，是在每一次试验下，要对论域中的一个固定元素ξ是否属于集合A'作出一个确切的判断。更深入的模糊统计试验要求在更深入的层次上作出确切的判断。这就要求，在每一次试验之下，A'必须是一个取定的普通集合。

做n次试验，计算ξ对模糊集合A的隶属频率：

$$\xi\text{对}A\text{的隶属频率}=\frac{\text{“}\xi\in A'\text{”的次数}}{n}$$

许多试验表明，随着n的增大，隶属频率也会呈现稳定性，称之为隶属频率的稳定性。频率稳定所在的数值，就称为元素ξ对模糊集合A的隶属度。

例如，有人曾做过抽样试验，取年龄论域为[0,100]，A为论域上表示“青年人”的模糊集。选取年龄$\xi=27$。用模糊统计来确定ξ对模糊集A的隶属度。选择129位合适的人选，在他们独自认真地考虑了“青年人”的含义之后，报出了他们认为的“青年人”的最适宜的年限，这个年限的上下限即为集合A'。如果“27岁”在这个上下限之内，则计一次“$\xi\in A'$”的次数。最后得到的结果是，27岁对于“青年人”年限的隶属频率大致稳定在0.78附近。因此，可以取

$$\mu_A(\xi)=\mu_A(27)=0.78$$

同样，按此方法不难求出“青年人”模糊集A的隶属函数。例如，将论域[0,100]分为m组，每组以中值为代表，统计每一组的隶属频率并将其作为该组属于集合A的隶属度。连续地描出其图形，即得到隶属函数$\mu_A(\xi)$的曲线。

2. 三分法

三分法也是用随机区间的思想处理模糊性的试验模型。例如建立“矮个子”A_1、“中等个子”A_2与“高个子”A_3这三个模糊集合的隶属函数。取论域为[0,3]（单位：米）。每一次模糊试验确定论域的一次划分，每次划分确定一对数(α,β)，α是矮个子与中等个子的分界点，β是中等个子与高个子的分界点。这样便把该模糊试验转化为如下的随机试验：

视(α,β)为二维随机变量，通过抽样调查，求得α与β的概率分布$p_\alpha(\xi)$与$p_\beta(\xi)$。则A_1，A_2，A_3的隶属函数为：

$$\mu_{A_1}(\xi) = \int_{\xi}^{-\infty} p_\alpha(\xi)\mathrm{d}\xi$$

$$\mu_{A_3}(\xi) = \int_{-\infty}^{\xi} p_\beta(\xi)\mathrm{d}\xi$$

$$\mu_{A_2}(\xi) = 1 - \mu_{A_1}(\xi) - \mu_{A_3}(\xi)$$

通常α与β都服从正态分布。设$\alpha: N(a_1,\sigma_1)$，$\beta: N(a_2,\sigma_2)$，则有

$$\mu_{A_1}(\xi) = 1 - \varphi\left(\frac{\xi - a_1}{\sigma_1}\right)$$

$$\mu_{A_3}(\xi) = \varphi\left(\frac{\xi - a_2}{\sigma_2}\right)$$

$$\mu_{A_2}(\xi) = \varphi\left(\frac{\xi - a_1}{\sigma_1}\right) - \varphi\left(\frac{\xi - a_2}{\sigma_2}\right)$$

其中，$\varphi(\xi) = \int_{-\infty}^{\xi} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}} \mathrm{d}x$

3. 增量法

同样，以前面关于年龄划分的例子来说明用“增量法”求隶属函数。例如求关于“老年”的模糊集A的隶属函数$\mu_A(\xi)$。任给ξ一个增量$\Delta\xi$，相应地μ有一个增量$\Delta\mu$。作为简化条件，可以认为$\Delta\mu$与$\Delta\xi$成正比。另一方面，对于同样大的增量$\Delta\xi$，若ξ越大，则$\Delta\mu$也应越大。再有，因为μ的值不能超过1，故当μ越接近1，$\Delta\mu$应越小。因此有

$$\Delta\mu = k \cdot \Delta\xi \cdot \xi(1 - \mu)$$

其中k为比例常数。将$\Delta\xi$移至左边，并令$\Delta\xi \to 0$，则得到微分方程：

$$\frac{\mathrm{d}\mu}{\mathrm{d}\xi} = k \cdot \xi(1 - \mu)$$

解此微分方程，有

$$\mu(\xi) = 1 - c\mathrm{e}^{\frac{k\xi^2}{2}}$$

其中c为积分常数。选择适当的k和c，即确定了$\mu(\xi)$，也就是确定隶属函数$\mu_A(\xi)$。

1.3.2 几种常用的隶属函数

1. B 样条隶属函数(B-spline membership functions)

B 样条基函数形式如下

$$\phi_{i,k_i,j}(\xi_i) = \left(\frac{\xi_i - \lambda_{i,j-k_i}}{\lambda_{i,j-1} - \lambda_{i,j-k_i}}\right)\phi_{i,k_i-1,j-1}(\xi_i) +$$

$$\left(\frac{\lambda_{i,j}-\xi_i}{\lambda_{i,j}-\lambda_{i,j-k_i+1}}\right)\phi_{i,k_i-1,j}(\xi_i)$$

$$\phi_{i,1,j}(\xi_i)=\begin{cases}1 & if\ \xi_i\in I_{i,j-1}\\ 0 & \text{其它}\end{cases}$$

B 样条基函数是一个简单的分段多项式映射，它被广泛地用于曲线拟合。关于 B 样条函数的详细情况请参见相关的书籍。B 样条函数可用于表示一个模糊隶属函数集合，隶属函数的形状由一组称为“节点向量”(knot vector)的参数决定。许多模糊设计者在使用三角形隶属函数实现其语言值时，隐含地使用了 2 阶 B 样条基函数，如图 1-5 所示。结点集$\{\lambda_i\}$决定了这些三角模糊集的宽度与位置(每一个惟一地定义在 $k+1$ 个连续值上)，并影响模糊系统的建模与学习能力。结点集形成一个简便的参数集合，以存储模糊集的定义，这在输出计算中是极为有用的。就如同一个简单的多项式可以用于决定对一个特定的输入而言，哪一个输入集是非零一样。

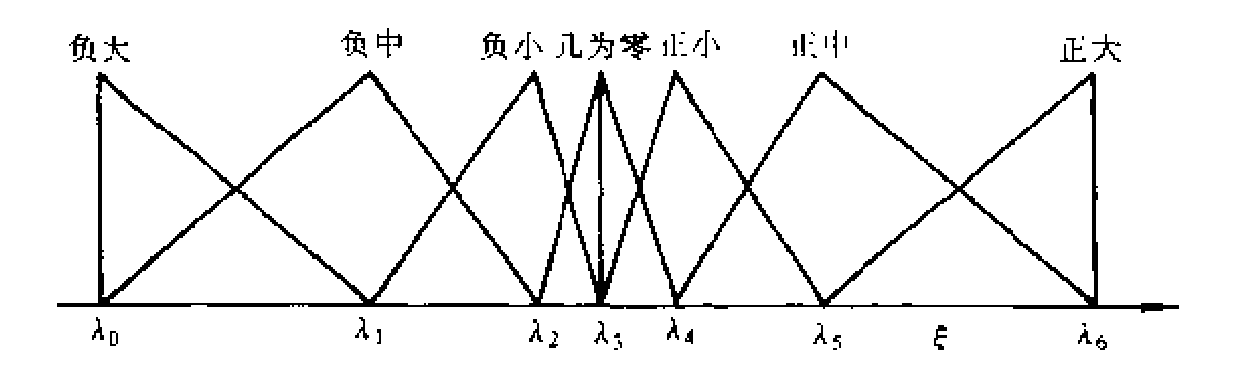


图 1-5　含有 7 个三角隶属函数语言值的模糊集合

更一般地，k 阶 B 样条基函数可以对一个明确的非模糊的表述($k=1$)建模，如果应用 3 阶或 4 阶 B 样条基函数时，则能生成更平滑的 2 阶或 3 阶映射，如图 1-6 和 1-7 所示。当用加法算于表示逻辑并(OR)，将集合连结到一起时，就形成普遍使用的梯形模糊隶属函数和 π 形模糊隶属函数。B 样条基函数 $\phi_{k,i}(\xi)$还具有如下的特性：

(1)它们是 k 阶分段多项式；

(2)它们定义在紧支集上，其输出只在 k 区间上是非零的；

(3)它们构成一个单位分解，即：$\sum_{i=1}^{k}\phi_{k,i}(\xi)=1$；

(4)它们的估计运算规则稳定且有效。

B 样条模糊隶属函数是分段多项式这一事实，意味着其应用极具灵活性，而且可以证明，一个基于 B 样条的模糊系统是泛逼近的，就是说，它们可以在一个紧域上任意逼近任何的连续非线性系统。紧支集性质意味着只有少数的模糊规则起作用当只有少数的隶属值非零时。它还将网络稳定度引入学习算法中，因为训练输入空间的一部分对于存储在一个不相似区域

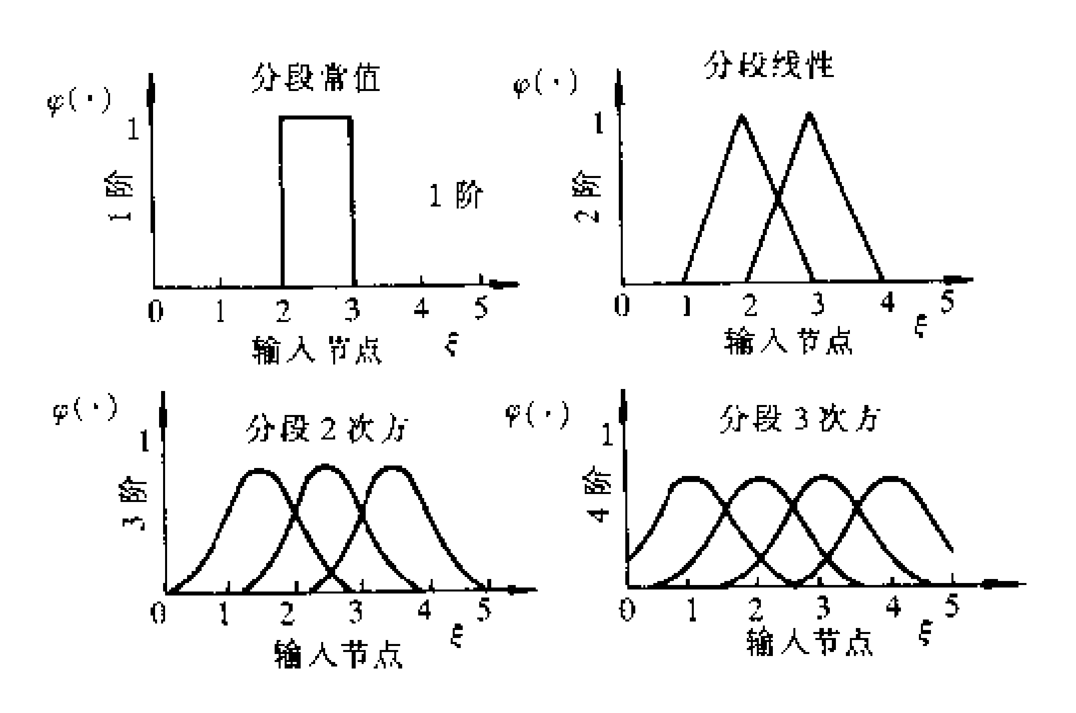


图 1-6　1 阶到 4 阶 B 样条函数

的知识没有显著的影响。一个单位分解的基函数集合是自标准化的。可以证明，模糊系统的反模糊化过程隐含地在网络上加上单位分解，这将显著地影响模糊隶属函数的构成。这种周期性的用以计算函数输出的关系如下式所示：

$$\phi_{i,1,j}(\xi_i) = \begin{cases} 1 & \xi_i \in I_{i,j-1} \\ 0 & \text{其它} \end{cases}$$

适当地选择节点的位置，就可以设计基函数，使其在期望函数快速变化的区域显著地改变。同时，在同一位置指定产生多个节点，就可以对数据中的不连续性建模。

普通 B 样条模糊隶属函数一个可能的限制，是其阶数与其支撑尺度相等。当要求较宽的基函数时，就意味着要使用高阶的、极为灵活的集合。当所要求的支撑尺度是一个基函数阶数的整数倍时，加宽的 B 样条减弱了这种关系。与普通 B 样条基函数一样，其加宽的相应部分应当满足前面提到的四个期望性质。

2. 高斯隶属函数

另一个常用来表示模糊隶属函数的集合是如下定义的高斯函数：

$$\mu_A(\xi) = \exp\left(-\frac{(c_i-\xi)^2}{2\sigma_i^2}\right)$$

高斯基函数易于实现，其中心(c_i)和宽度(σ_i)构成了一个简便的参数集合以进行初始化(并可用于自适应模糊系统的训练)。适当地选择中心和宽度，则高斯函数也同样极具灵活性。与 sigmoidal 型(sigmoidal-type)映射一样，网络输出可以逼近局部线性函数。高斯函数也是局部定义的，但严格地说它们不具有紧支集。在其它文献中提出过具有紧支集的修正高斯型隶属函数，其定义如下：

$$\mu_{N'}(\xi)=\begin{cases}\exp\left(-\dfrac{(\lambda_2-\lambda_1)^2/4}{(\xi-\lambda_1)(\lambda_2-\xi)}\right) & \xi\in(\lambda_1,\lambda_2)\\ 0 & \text{其它}\end{cases}$$

当模糊集的支撑为(λ_1,λ_2)时，函数在$\xi=(\lambda_1+\lambda_2)/2$处具有最大值，此时其值为$\exp(-1)$，如图1-7所示。

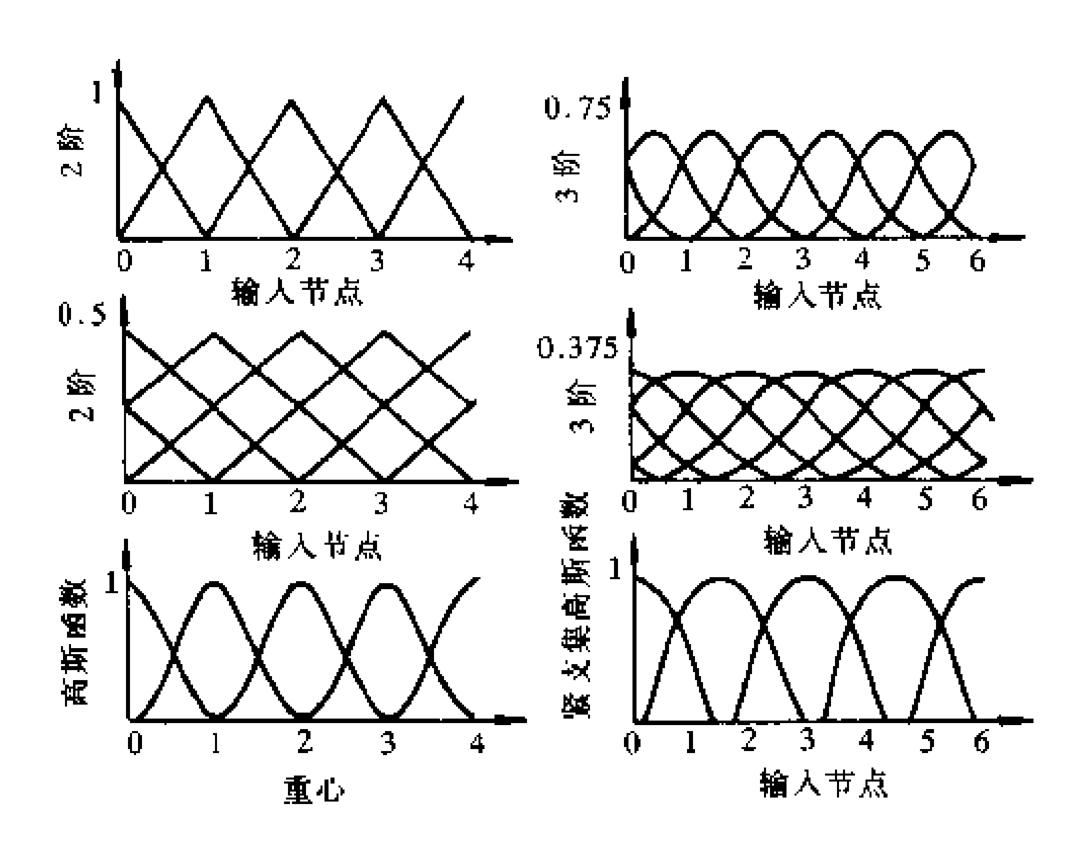


图1-7　6种不同的连续隶属函数(上部为B样条隶属函数，中部为扩展B样条隶属函数，底部为高斯和紧支集高斯隶属函数)

由图1-7可见，存在许多看起来不同的模糊集形式，且每一个模糊集都由一组参数决定集合的中心和宽度。

在数学上，两个最常用的隶属函数是B样条函数和高斯函数；而在工程应用中，其计算和推导显得过于麻烦。为了达到设计简便及实时计算的要求，在工程中往往采用形式上更简单的隶属函数，其中使用最多的是三角隶属函数。在MATLAB模糊逻辑工具箱中，内置了11种隶属函数以方便用户。下面对MATLAB模糊逻辑工具箱内置的隶属函数作一简单介绍。

1.3.3　模糊逻辑工具箱内置的隶属函数

模糊逻辑工具箱中包含了11个内置的隶属函数类型。这11个函数又由几个基函数构成：分段线性函数，高斯分布函数，S形曲线，二次和三次多项式曲线。关于下面介绍的隶属函数的详细情况，请查阅附录“模糊逻辑工具箱函数详解”。按照MATLAB的惯例，所有隶属函数最后的两个字母都是“mf”。

最简单的隶属函数是由直线构成的，这些直线隶属函数的优点之一就是简洁。其中最简单的是三角隶属函数，其函数名为 trimf，它是由三个点构成一个三角形，如图 1－8 所示。

另一个直线隶属函数是梯形隶属函数。梯形隶属函数 trapmf 有一个平顶，它是一个截去顶部的三角形曲线，如图 1－9 所示。

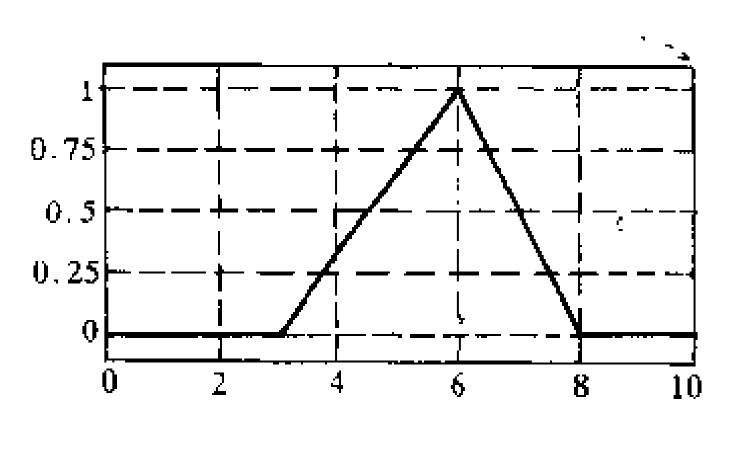


图 1－8 三角形隶属函数

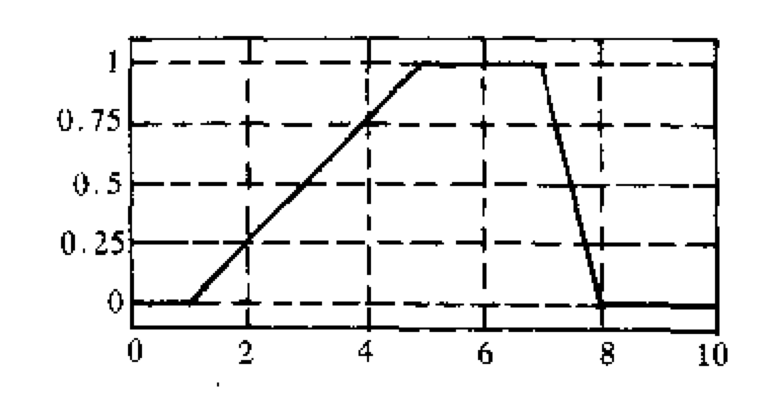


图 1－9 梯形隶属函数

有两个隶属函数建立在高斯分布曲线之上。一个是简单的高斯曲线，如图 1－10 所示；一个是曲线两侧由不同的高斯曲线组合而成，如图 1－11 所示。这两个函数是 gaussmf 和 gauss2mf。广义钟形隶属函数由 3 个参数指定，其函数名为 gbellmf。钟形隶属函数比高斯隶属函数多一个参数，因此只要适当地调整其参数，它就可以逼近一个非模糊集合，其曲线如图 1－12 所示。由于其平滑性以及表示的简洁性，高斯隶属函数和钟形隶属函数也是目前使用最广泛的定义模糊集合的方法之一。这两个隶属函数的优点是其曲线在所有点上都平滑且非零。

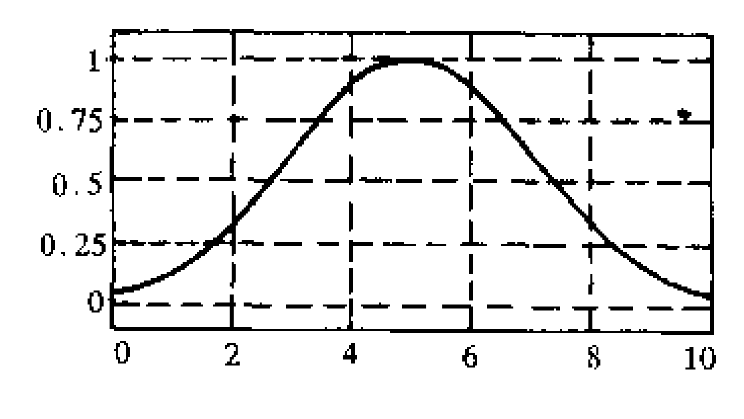


图 1－10 简单高斯隶属函数

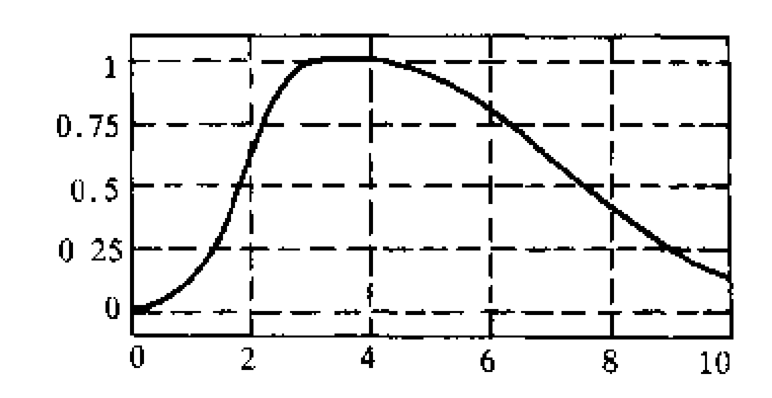


图 1－11 复合高斯隶属函数

尽管高斯隶属函数和钟形隶属函数具有平滑性，但它们不能规定非对称的隶属函数，而这在某些应用中是相当重要的。因此模糊逻辑工具箱中内置了 sigmoidal 隶属函数（sigmoidal MF），它是左开或右开的。非对称的或封闭的（左闭或右闭）的隶属函数可以由两个 S 形函数结合构成。故除了基本的 sigmf 函数（即简单 sigmoidal 隶属函数，如图 1－13 所示），还有两

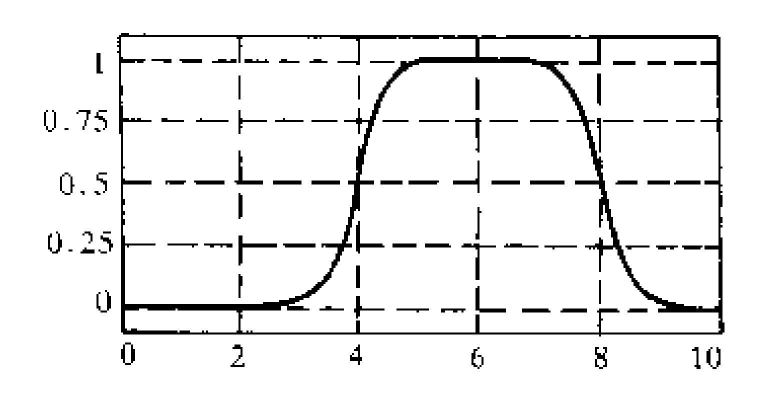


图1-12　钟形隶属函数

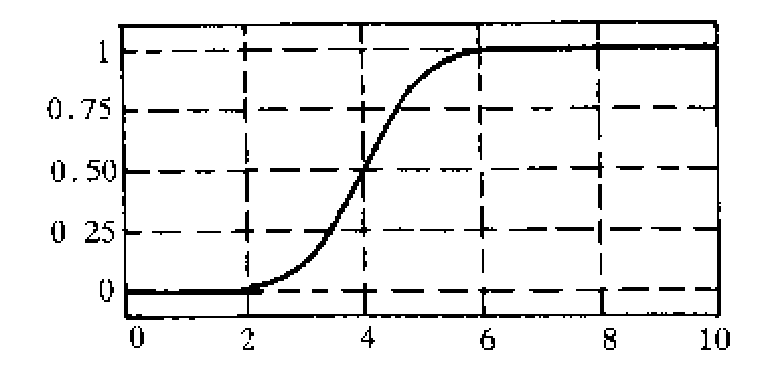


图1-13　简单sigmoidal隶属函数

个S函数之差dsigmf函数(即差型sigmoidal隶属函数,如图1-14所示),以及两个S函数之积的psigmf函数(即积型sigmoidal隶属函数,如图1-15所示)。

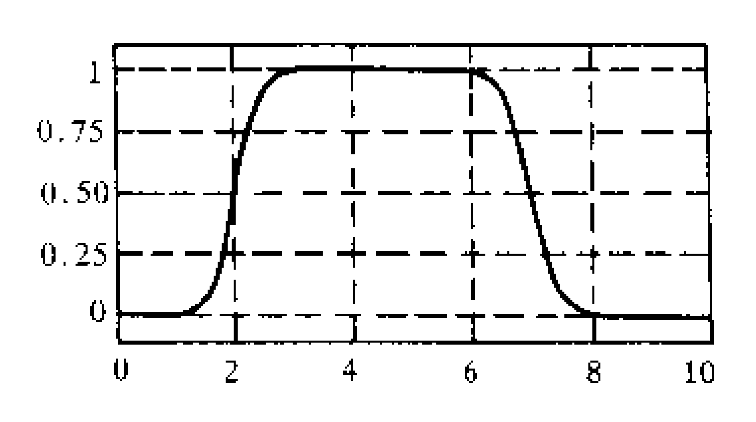


图1-14　差型sigmoidal隶属函数

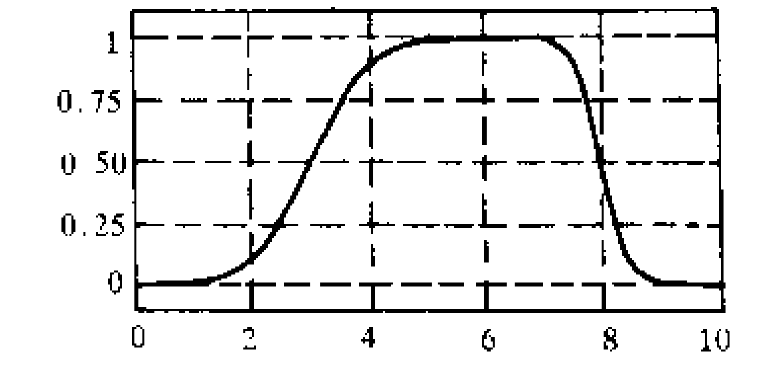


图1-15　积型sigmoidal隶属函数

在工具箱中有若干个基于多项式曲线的隶属函数。三个相关的隶属函数是Z(即Z形隶属函数,如图1-16所示),S(即S形隶属函数,如图1-17所示)以及Pi(即π形隶属函数,如图1-18所示)曲线,它们的名字与其曲线形状相对应。函数zmf是左开的非对称多项式曲线,smf是右开的镜像函数,pimf是在两端为0中部上升的曲线。

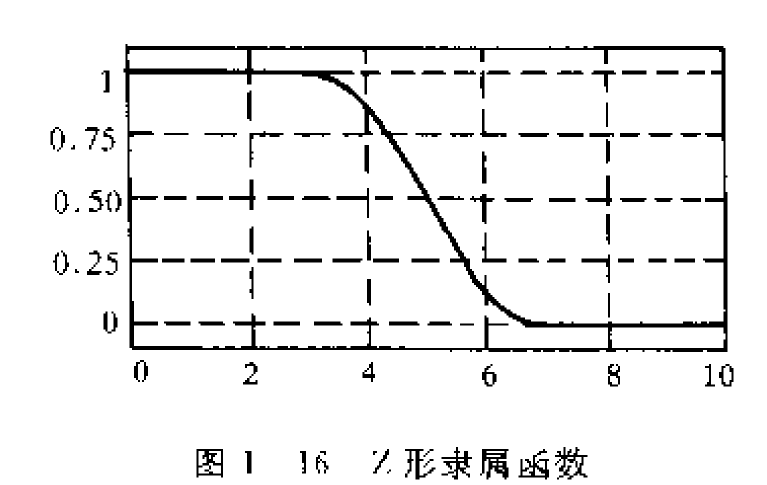


图1-16　Z形隶属函数

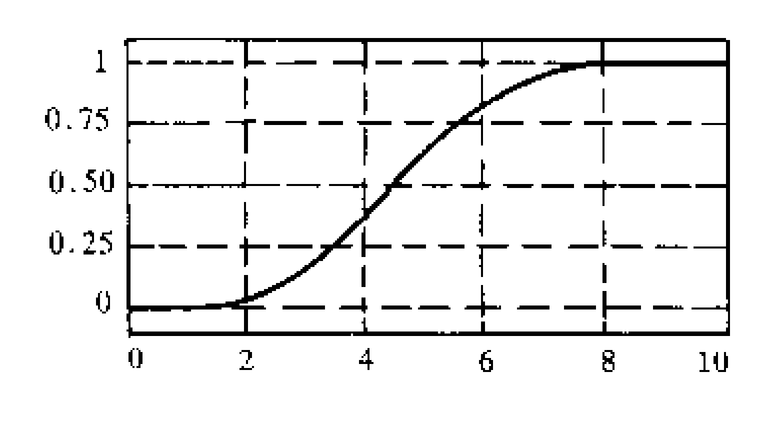


图1-17　S形隶属函数

在模糊逻辑工具箱选择所需的隶属函数时,由于模糊逻辑工具箱内置了诸多的隶属函数,因此有很大的选择范围。并且,用户如果觉得选择列表中的隶属函数有局限时,还可以用模糊逻辑工具箱建立自己的隶属函数。由于选择列表中的隶属函数太多,因此只需要记住其中的

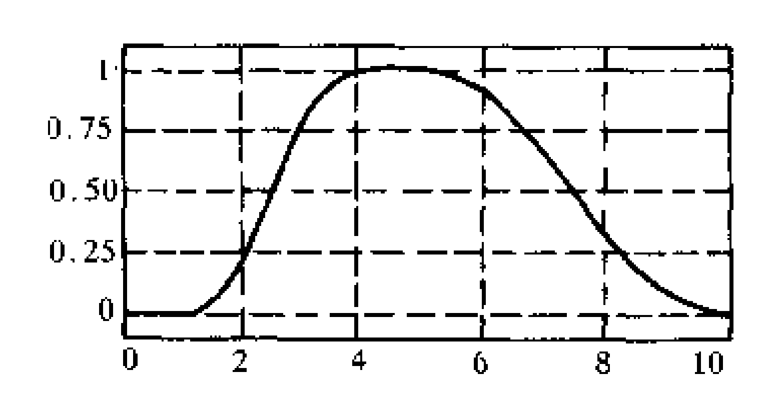


图 1-18　π 形隶属函数

一到两种隶属函数就可以了，比如三角函数和梯形函数。一般而言，模糊逻辑工具箱内置的隶属函数已经足够使用。但要完成一个完善的模糊推理系统，还是有必要建立自己的隶属函数。

1.4　模糊运算与模糊推理

1.4.1　模糊运算

与经典集合的并、交、补的运算相对应，模糊集合也有相似的运算。首先我们介绍最简单的模糊并、模糊交与模糊补的概念。对于一般性的理解，这样的介绍已经足够了。在其后我们将更深入地介绍其更一般的意义。

1. 模糊子集

当且仅当对所有的 ξ，均有 $\mu_A(\xi)\leqslant\mu_B(\xi)$，则称模糊集合 A 被包含在模糊集合 B 中，或称 A 是 B 的子集，或称模糊集合 A 小于或等于模糊集合 B。记为

$$A\subseteq B \leftrightarrow \mu_A(\xi)\leqslant\mu_B(\xi)$$

例如，模糊控制器的语言变量是指其输入变量和输出变量，在输入变量或输出变量的论域上，往往需要为语言变量选取多个语言变量值（以后均按惯例统称为语言值），如“正大”、“正中”、“正小”、“几为零”、“负大”、“负中”、“负小”等，它们就分别是一个模糊子集。又如，以人们通常概念上（而不是法定意义的“大于 18 周岁”）的“成年人”作为一个模糊集合，那么就可以用“青年人”、“壮年人”、“中年人”和“老年人”作为“成年人”的语言值。而这几个语言值都分别是“成年人”模糊集合的模糊子集。模糊子集的范畴比语言值更广，但人们使用更多的是模糊集合的语言值。

2. 模糊并（一）

两个模糊集合 A 和 B 的“并”为模糊集合 C。写成 $C=A\cup B$，或 $C=A\ or\ B$。C 与 A 和 B 的隶属函数的关系为

$$\mu_C(\xi)=\max(\mu_A(\xi),\ \mu_B(\xi))$$

显然，由模糊子集的关系，可以很容易地理解模糊并的算子为 max、即两者取其大。

3. 模糊交(一)

两个模糊集合 A 和 B 的"交"为模糊集合 C。写成 $C=A\cap B$、或 $C=A$ and B。C 与 A 和 B 的隶属函数的关系为

$$\mu_C(\xi) = \min(\mu_A(\xi), \mu_B(\xi))$$

同样，由模糊子集的关系，可以很容易地理解模糊交的算子为 min，即两者取其小。

4. 模糊补

模糊集合 A 的补表示为 $-A$ 或 $\overline{A}$(非 A)。模糊补的隶属函数定义为

$$\mu_{-A}(\xi) = 1 - \mu_A(\xi)$$

图 1－19，图 1－20，图 1－21，图 1－22 分别给出了模糊集合 A 和 B，$A\cup B$，$A\cap B$ 和 $-B$ 的图形。

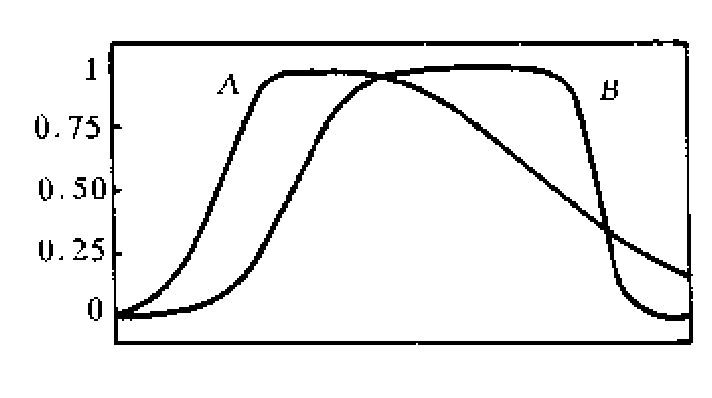


图 1－19　两个模糊集 A 与 B

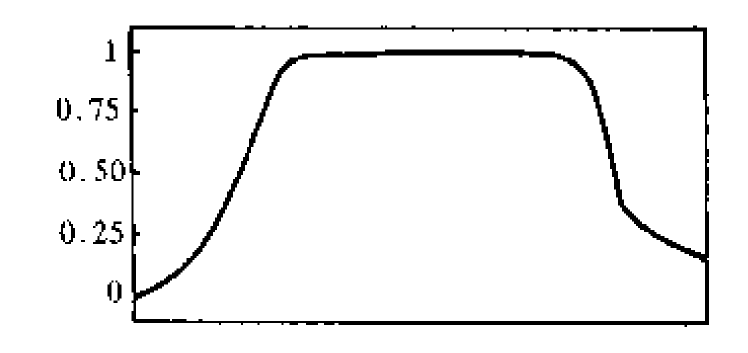


图 1－20　$A\cup B$

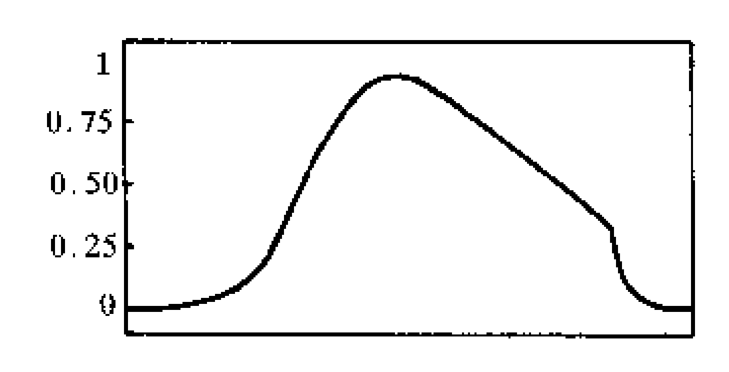


图 1－21　$A\cap B$

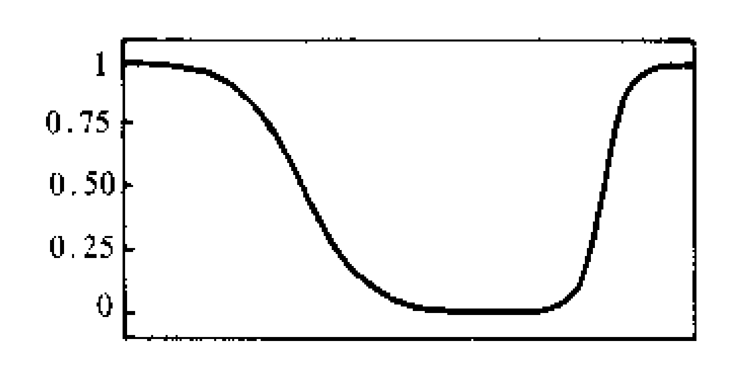


图 1－22　$-B$

一般地，模糊集合 A 和 B 的交集可由一个二元映射 T 来指定，它将两个隶属函数按如下方式结合起来

$$\mu_{A\cap B}(\xi) = T(\mu_A(\xi), \mu_B(\xi))$$

例如，二元运算 T 可以代表 $\mu_A(\xi)$ 和 $\mu_B(\xi)$ 的乘积。这些模糊交集算子通常被归为 T 范式(三角范式)算子，它们应当满足如下的基本要求。

如果一个 T 范式算子是一个二元映射 $T(\cdot,\cdot)$，则应当满足：

有界性　$T(0,0)=0, T(a,1)=T(1,a)=a$

单调性　$T(a,b)\leqslant T(c,d)$, if $a\leqslant c$ and $b\leqslant d$

交换律　$T(a,b)=T(b,a)$

结合律　$T(a,T(b,c))=T(T(a,b),c)$

第一个要求使得一个明确的集合具有适当的一般性；第二个要求意味着集合 A 或 B 中隶属值的减少不会导致 A，B 交集中隶属度的增加；第三个要求表明运算符对模糊集合的顺序是无关紧要的；第四个要求使我们可以按任意的顺序对任意多个集合进行模糊交运算。

与模糊交一样，一般地，模糊并的运算由一个指定的二元映射 S 产生

$$\mu_{A\cup B}(\xi)=S(\mu_A(\xi),\mu_B(\xi))$$

例如，二元运算 S 可以表示 $\mu_A(\xi)$ 和 $\mu_B(\xi)$ 的加法。这些模糊并算子通常被归为 T 协范式（或 S 范式）运算符，它必须满足下列基本要求。

T 协范式（或 S 范式）算子是一个二元映射 $S(\cdot,\cdot)$，则它必须满足：

有界性　$S(1,1)=1, S(a,0)=S(0,a)=a$

单调性　$S(a,b)\leqslant S(c,d)$, if $a\leqslant c$ and $b\leqslant d$

交换律　$S(a,b)=S(b,a)$

结合律　$S(a,S(b,c))=S(S(a,b),c)$

在过去，人们已经提出了多参数的 T 范式和双 T 协范式。其中每一种都提供了一种在函数中改变增益的方法，因此它可以具有很强的限制性，也可以具有很强的容许性。

5. 模糊交（二）

模糊乘规则的前提是由 n 个单变量语句的模糊交（AND）构成

$$(\xi_1 \text{ is } \mu_{A_1^i}) \text{ AND } \cdots \text{ AND } (\xi_n \text{ is } \mu_{A_n^i})$$

它产生一个新的多变量隶属函数 $\mu_{A_1^i\cap\cdots\cap A_n^i}(\xi_1,\cdots,\xi_n)$，记为 $\mu_{A^i}(\xi)$，它定义在初始的 n 维输入空间上，其输出如下：

$$\mu_{A^i}(\xi)=\overset{\wedge}{\prod}(\mu_{A_1^i}(\xi_1),\cdots,\mu_{A_n^i}(\xi_n))$$

其中 $\overset{\wedge}{\prod}$ 是一类称为三角范式（triangular norm）的函数。三角范式提供了大量函数以实现模糊交，其中最常用的两个函数是 min 算子和乘法（product）算子。一个二维的模糊隶属函数由两个三角（2 阶 B 样条）单变量隶属函数的乘积构成，其形状如图 1－23 所示。

显然，多变量隶属函数的形状决定于单变量隶属函数的形状以及三角范式的算子。用乘法算子构成的多变量隶属函数所保留的信息，比用 min 算子实现模糊“AND”时所保留的信息要多。因为后者仅保留了一段信息，而乘法算子结合了 n 段信息。当完整地定义了一阶导数时，采用乘法算子还允许将误差信息反向传播回网络中。一般情况下，其输出结果是一个更平滑的曲面。当每一个语句表达式都用单变量 B 样条和高斯模糊隶属函数表示时，多变量隶属

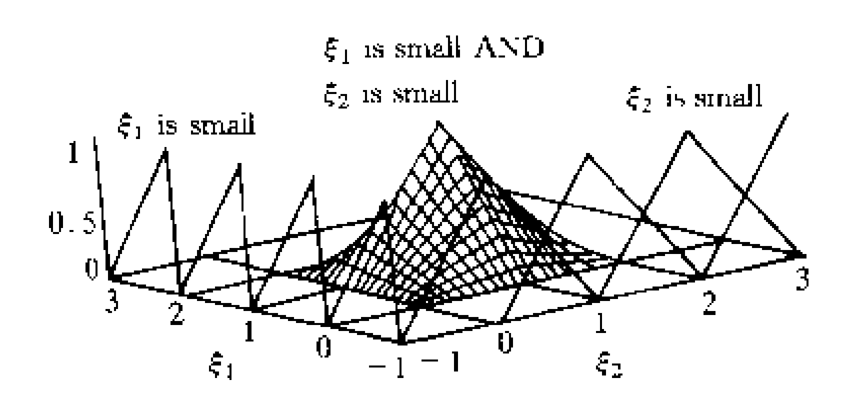


图 1－23　两个三角单变量隶属函数乘积构成的二维模糊隶属函数

函数是一个简单的 n 维 B 样条或高斯基函数。

当所有可能的模糊交集都由 n 个模糊隶属函数集合得到时，它隐含地在初始输入空间上（多变量模糊隶属函数也在这个初始输入空间上定义）产生一个 n 维网格。如图 1－24 所示。

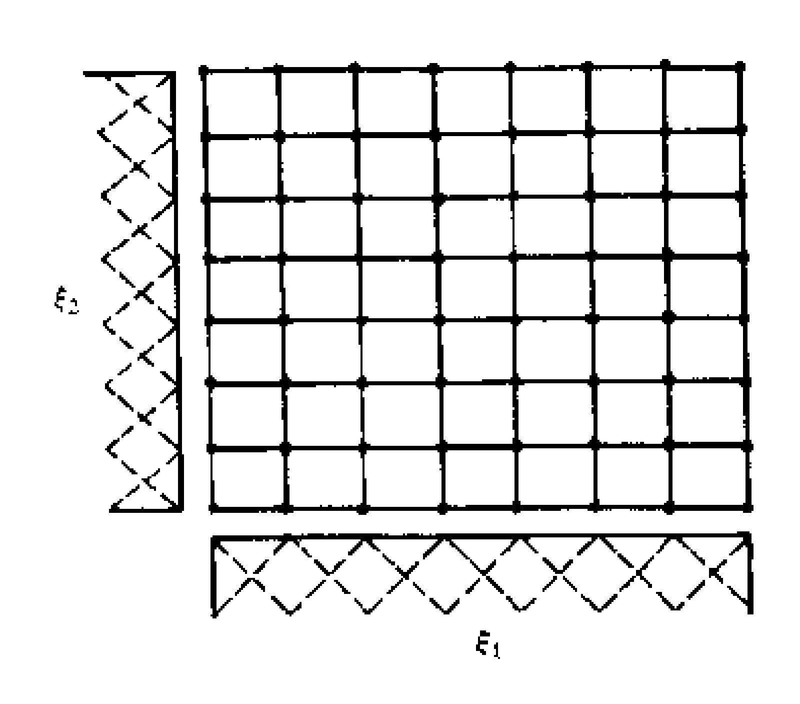


图 1－24　由两个三角模糊集构成的二维模糊集

其中一个完整的二维模糊隶属函数集合由两个三角单变量模糊集合产生。图中的实心圆表示其中心，虚线区表示两个单变量集合如何用交集算子结合起来。当模糊交集由每一个可能的单变量模糊输入集的结合得到时，多变量隶属函数的数目是输入变量数目的指数函数（exponential function）。称这种模糊系统是完备的。因为对每一个输入，至少存在一个具有非 0 隶属度的多变量模糊集。只要从图 1－24 中移去一个多变量模糊集，则意味着模糊规则库不再是完备的（因为在移去集合的中心，每个基函数的隶属度为 0）。假设一个系统有 4 个输入变量，一个输出变量，每个变量各有 7 个语言值，则一个完备的模糊网络将会有 7^5 = 16 807个中心点。因此，除非用特定的方法构造模糊网络的输入，否则这些系统将会遇到所谓

的"维数灾难(curse of dimensionality)",这限制了它们用于小维数的建模与控制问题。

6. 模糊并(二)

如果 h 个多变量模糊输入集合 A^i 映射到 q 个单变量模糊输出集合 B^j 上,则对 hq 个逻辑关系,存在 hq 个相对应的叠加的 $n+1$ 维隶属函数。这 hq 个逻辑关系由此被各个隶属函数并(OR)相结合,构成一个模糊规则库 R。该运算定义如下

$$\mu_R(\xi,y)=\hat{\sum_{i,j}}\mu_{r_{i,j}}(\xi,y)$$

其中 $\hat{\sum}$ 为一类称为三角协范式的函数。三角协范式同样提供了大量的适用函数,但使用最多的两个函数是 max 算子和加法算子。用 max 算子可以保证在输入输出空间每一特定的点上,只有一个规则对输出有作用。而各个作用之和可以保证若干规则影响相关曲面(relational surface)$\mu_R(\xi,y)$。但是,在理论上用加法不能保证合理性,因为它产生的输出可能大于 1。当输入输出单变量隶属函数构成单位分解、乘法算子用于交集和蕴涵,并且规则信度向量归一时,这就不是什么问题了,因为系统是自标准的。相似地,当用不同的反模糊化过程执行蕴涵标准化时,在蕴涵中同样可以使用加法算子,即使它产生的 $\mu_R(\xi,y)$ 的值大于 1。

所有各相关隶属函数的模糊并,在输入输出空间中构成一条岭线,它表示各输入输出对是如何相联系的,并且当给定一个特定的输入测量时,它可以用于推断模糊输出隶属函数。这个过程称为模糊推理。一个典型的模糊相关关系如图 1-25 所示。

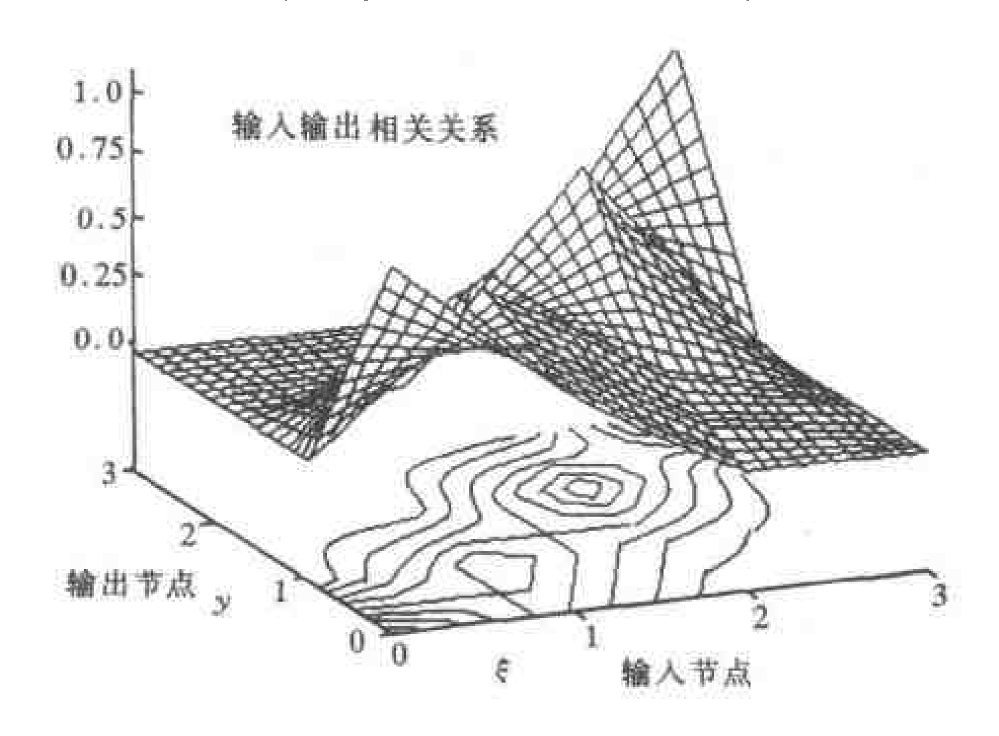


图 1-25　模糊输入输出相关关系

其中,在每一个变量上定义了 4 个三角模糊集(2 阶 B 样条),代数函数用于实现逻辑运算。它们产生一个模糊相关关系曲面,在规则中心点之间是分段线性的,而且从等高线图可以清晰地看出输入和输出关系的总趋势。

1.4.2 模糊规则与模糊推理

在前面的描述中，我们已经提到了模糊推理和模糊规则的概念。在模糊推理中，经常碰到模糊 if-then 规则，简称模糊规则。我们首先来理解模糊规则的概念。

1. 模糊规则

模糊 if-then 规则又称模糊隐含或模糊条件语句。

if-then 规则语句用以阐明包含模糊逻辑的条件语句。一个单独的模糊 if-then 规则形式如下：

if x is A then y is B

其中，A 和 B 是由模糊集合分别定义在 X，Y 范围(论域)上的语言值。模糊规则中的 if 部分"x is A"被称为规则的前提或假设，同时 then 部分"y is B"被称为结果或结论。实质上，该表达式描述了变量 x 与 y 之间的关系。因此，我们可以把 if-then 规则定义为乘积空间中的二元模糊关系。

例如，要购买一个软件，其价格由其用户界面和软件功能决定。若单独考虑其价格，则

if interface is good then charge is high

注意"good"用一个 0 和 1 之间的数字表示，因此所谓的前提是一个解析，它返回一个 0 和 1 之间的单值。另一面，"high"由一个模糊集合表示，因此所谓的结果是一个分配，它分配整个模糊集合 B 到输出变量 y。在 if-then 规则中，当"is"分别出现在前提和结果中时，其意义完全不同。就如在 MATLAB 术语中，使用关系运算符"=="和使用变量赋值符号"="时，其意义也完全不同。

书写这个规则时，避免混淆的写法是

if interface==good then charge=high

一般地，if-then 规则的输入是输入变量的当前值(在此是"interface")，输出是一个模糊集合的整体(在此是"high")。在后面将对这个集合进行反模糊化，将一个值分配到输出。反模糊化的概念将在后面描述。

在模糊推理中，一个规则的前提可以有多个部分。例如

if interface is good and performance is good then charge is high

在这种情况下，前提的所有部分都同时使用前面描述的逻辑算子进行计算，并分配为一个单值。

规则的结果同样也有多重部分：

if interface is bad then charge is low and cash in order is low

在这种情况下，所有的结果都同等地被前提影响。面前提是怎样影响结果的呢？在模糊逻辑中，模糊推理的结果将一个模糊集合分配到输出中，然后模糊规则按前提中指定的程度修改模

糊集合。最常用的修改输出模糊集的方法，是用 min 函数进行截断或用 prod 函数进行缩放。

对于更多个变量的输入输出关系，令一个规则将第 i 个多变量模糊输入集合 A^i 映射到第 j 个单变量输出集合 B^j 上，并以 c_{ij} 表示其信度，这种关系称为模糊蕴涵或简称蕴涵，例如

$$r_{ij}: \text{IF}(\xi \text{ is } A^i) \text{ THEN } (y \text{ is } B^j)\ (c_{ij})$$

则元素 x 和元素 y 的相关程度用一个定义在乘法空间 $A_1 \times \cdots \times A_n \times B$ 上的 $n+1$ 维隶属函数表示

$$\mu_{r_{ij}}(\xi, y) = \overset{\wedge}{*} \mu_{A^i}(\xi) \overset{\wedge}{*} c_{ij} \overset{\wedge}{*} \mu_{B^j}(y)$$

其中 $\overset{\wedge}{*}$ 为二元三角范式，通常用 min 算子或乘法算子。当第 ij 个模糊规则的输入为 ξ 时，模糊集 $\mu_{r_{ij}}(\xi, y)$ 表示输出为 y 的信度。

在这些应用中，模糊蕴涵可以认为是输入集合和输出集合的一个交集。这种解释不是唯一的，但是在大量的模糊建模和控制系统中都采用这种解释。

2. 模糊规则的信度

模糊推理系统的知识库中包括了模糊隶属函数的定义、模糊逻辑算子和 $(h \times q)$ 模糊规则信度矩阵(rule confidences matrix)$\boldsymbol{C}$。在模糊规则可信度矩阵 $\boldsymbol{C}$ 中，h 为多变量模糊输入集的个数，q 为单变量模糊输出集的个数。规则信度矩阵中的每一个元素 c_{ij} $(\in [0,1])$ 表示第 i 个多变量模糊输入集合与第 j 个单变量模糊输出集合相关的强度或称信度。当某个规则信度为 0，则表示输出集合对特定的模糊输入集合的输出没有作用。另一面，当一个规则的信度大于 0，则无论何时，只要输入部分地满足规则的前提，输出集合就将影响系统输出。因此，模糊规则的信度与模糊规则的权之间有着密切的关系，后面将对此作出解释。一旦定义了一个模糊集合，则规则信度就已封装一个特定过程的专家知识，它们形成一个简便的用于训练的参数集合。

规则信度与模糊集合的形状或类型和模糊逻辑算子无关。模糊集合和模糊逻辑算子分别独立地存储于知识库中。在自组织控制器中广泛使用的离散模糊系统构造了一个相关矩阵，它完整地表征了系统的知识库，并隐含地包括了模糊集的形状、逻辑算子和规则信度的信息。人们希望将模糊知识按前面描述过的分布形式存储起来，这将使得人们易于理解不同的实现方法是如何影响系统输出的。

规则信度向量 $\boldsymbol{c}_i$ 与每个多变量模糊输入集合相关联，它表征对特定输入集合的系统输出的估计。一般地，规则信度向量是标准的(归一的)，它表示，对于一个特定的输入集合，存在关于系统输出的全部知识。当规则库中的知识发生变化时，这些参数易于更新。在许多自适应模糊系统中，更改模糊输出隶属函数的方法是转移其中心，它等于重新定义设计者对语句表述的主观解释。因此，有理由提出，在完成训练之后，由于模糊集的形式与其初始定义不一致，因此不能认为这些自适应模糊系统是有效的。但是，当独立地使用和存储规则信度时，适应一个

规则作用的强度并重新获得其初始的模糊语义解释是可能的。

3. 模糊推理

建立了模糊规则(模糊蕴涵)的概念之后,我们来进行模糊推理的讨论。

推理是对于一个特定表述的解释过程,它利用所有的有用知识以产生最佳的输出估计。在模糊系统中,利用推理机制完成当前模糊输入集 $\mu_A(\xi)$ 与所有模糊规则前提的模式匹配,并结合其响应,产生一个单独的模糊输出集 $\mu_B(y)$。这个过程定义如下:

$$\mu_B(y) = \hat{\sum_\xi}(\mu_A(\xi) \hat{*} \mu_R(\xi, y))$$

其中,对于所有可能的 ξ 值,都应用三角协范式 $\hat{\sum}_\xi$。应用三角范式的目的,是计算对一个特定的 ξ 值,两个隶属函数之间的匹配。当 $\hat{\sum}$ 和 $\hat{*}$ 被分别用作积分(和)算子与乘法算子时,则

$$\mu_B(y) = \int_D \mu_A(\xi)\mu_R(\xi, y)\mathrm{d}\xi$$

上式要求对任意模糊输入集合,在输入域 D 上是 n 维可积的。对模糊输出集的计算依赖于模糊输入集 $\mu_A(\xi)$、相关曲面 $\mu_R(\cdot)$ 以及实际的推理算子。

只要在模糊输入集和规则库前提之间存在重叠,则在某种意义上,模糊系统具有归纳(generalise)能力。对相邻表述的信息进行归纳是模糊逻辑的功能之一。在本节中研究的模糊系统是相当重要的,因为对它的逼近能力能同时进行理论上的分析与决策。这对于实际系统具有重要的意义。

一般地,模糊推理可以分为4步:即计算隶属度,将已知事实与模糊规则的前提进行比较,求出相对每一前提隶属函数的隶属度;求激励强度,或称求总前提的满足程度,用模糊并或模糊交算子,把相对于前提隶属函数的隶属度结合起来,求出对总前提的满足程度;应用模糊规则,将激励强度施加于模糊规则结果的隶属函数,以产生一个定性的隶属函数;进行模糊聚类,获得最终输出的隶属度。

模糊推理的输出结果是一个模糊集,而模糊控制器的输出必须是一个确定的数值。这就是涉及到推理结果的反模糊化问题。通常对模糊推理结果有下面几种反模糊化方法。

(1)简单平均法

取 $y(\xi)$ 为

$$y(\xi) = \frac{1}{2}(\inf A_\xi + \sup A_\xi)$$

(2)最大隶属函数法

选择 y 使

$$y(\xi) = \max_{y \in Y} \mu(y)$$

用上式确定的 y 有时不唯一。对此问题的解决方法是，取使上式成立的多个结果的平均值作为 $y(\xi)$。

(3)重力中心法

$$y(\xi)=\frac{\int_Y \mu_B(y)y\mathrm{d}y}{\int \mu_B(y)\mathrm{d}y}$$

在此要求分子与分母的积分都存在。重力中心法考虑到了 $\mu_B(y)$ 的形状，采用了较多的信息，是比较常用的方法。

(4)α 水平重力中心法

$$y(\xi)=\frac{\int_{J_\alpha} \mu_B(y)y\mathrm{d}y}{\int_{J_\alpha} \mu_B(y)\mathrm{d}y}$$

1.4.3 Mamdani 型推理与 Sugeno 型推理

到目前为止，我们所讨论的模糊推理过程都是 Mamdani 模糊推理方法。本书内容都是在采用 Mamdani 推理的基础上进行的。Mamdani 推理方法是使用最多，同时也比较简便的模糊推理方法。但在文献中时常见到使用 Sugeno 型模糊推理的例子，在此对其做一简单介绍。

Sugeno 方法又称为 Takagi-Sugeno-Kang 方法，它于 1985 年首次提出。它在许多方面与 Mamdani 方法是相似的。事实上，在模糊推理进程的前两个部分，即输入模糊化和应用模糊算子，两者是完全相同的。

在 Sugeno 型模糊推理和 Mamdani 型模糊推理之间主要的不同是，对 Sugeno 型模糊推理，输出隶属函数只能是线性的或者是常量。

如前所述，Mamdani 型推理要求输出的隶属函数为一个模糊集。在完成聚类过程之后，对每个需要被反模糊化的输出变量，存在一个模糊集合。而在许多情况下，使用一个模糊单点而不是一个分布模糊集作为输出隶属函数更为有效。

模糊单点的数学表示如下

$$\mu_A(\xi)=\begin{cases}1 & \xi=\xi' \\ 0 & \text{其它}\end{cases}$$

图 1－26 中表征了身高为“1.6 米”的模糊单点。

这个模糊单点被称为单元输出隶属函数，它可以被视为一个预反模糊化集合。它极大地降低了更一般的、找到二维函数重心的 Mamdani 方法所需的计算量，因此提高了反模糊化过程的效率。

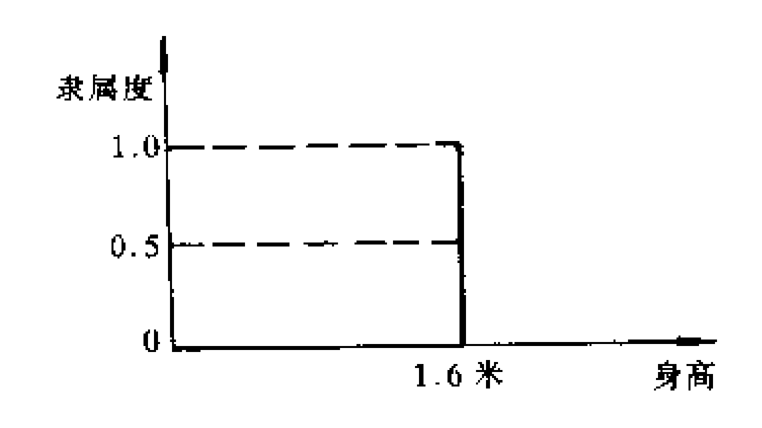


图 1-26　模糊单点

Sugeno 型系统正是支持这一类模型的推理系统。在寻找二维函数重心的过程中，我们使用的是一些数据点的加权平均值，而不是对二维函数进行积分。在零阶 Sugeno 模糊模型中，一个典型的模糊规则形式如下：

if x is A and y is B then $z = k$

其中 A 和 B 是前提中的模糊集，k 是结果中具有明确定义的常量。当 Sugeno 型模糊推理每一规则的输出都是这种常量时，它与 Mamdani 方法的相似性达到惊人的程度。

一般说来，可以使用 Sugeno 型系统对任何推理系统建模，无论推理系统的输出隶属函数是线性的还是常值。

要想象一个一阶 Sugeno 型系统，最容易的方法是设想将每个规则的位置定义为一个“移动的单点”，即，在输出空间中模糊单点输出函数能以一种线性方式移动，而移动的方式取决于输入。这将使系统的表示更紧凑而有效。更高阶的 Sugeno 模糊模型也是可能的，但它们具有极大的复杂性，而缺乏明显的优点，因此其应用数量相当少，而 MATLAB 模糊逻辑工具箱也不支持输出隶属函数高于一阶的 Sugeno 模糊模型。

由于在系统输入变量上每个规则线性相关，对于分别应用于一个动态非线性系统中不同条件下的复合线性控制器，Sugeno 方法是一个理想的内插方法。例如，飞机性能可能随着飞行高度和马赫数的变化而剧烈地改变。虽然在给定的条件下，线性控制器易于计算并性能良好，但在飞行状态不断变化时，它必须被有规律地、平滑地刷新。要平滑地插入在整个输入空间中应用的线性增益，Sugeno 模糊推理系统是最适合的。它是一个自然有效的增益序列。同样，Sugeno 系统也适于用内插复合线性模型对非线性系统建模。

Sugeno 型系统所得到的输出曲面与 Mamdani 系统的结果几乎完全一样。但由于 Sugeno 系统比 Mamdani 系统更紧凑，并具有更高的计算效率，所以它在建立模糊模型中可以使用自适应技术。这些自适应技术能用于自定义隶属函数，因此模糊系统能对数据完成最优建模。

Mamdani 方法的优点主要在于：直观，具有广泛的接受性以及尤其适于人工输入等。而 Sugeno 方法的主要优点在于计算效率高，使用线性化技术工作时性能良好（例如，PID 控制），

使用最优化和自适应工作时性能良好，能保证输出曲面的连续性以及尤其适于数学分析等。

1.5 模糊系统

1.5.1 模糊系统的结构

通常，在一个实际的模糊控制系统中，模糊推理系统的功能与模糊控制器的功能是等价的。从系统的观点而言，模糊控制器本身也就是一个系统。同时，在用 MATLAB 研究模糊控制系统时，Simulink 模糊控制仿真系统中的模糊控制器就是直接利用模糊逻辑工具箱建立的模糊推理系统。因此，在本书中，模糊控制系统与模糊控制器两个词的意义是相同的，两者可以互换使用。

一个模糊推理系统(fuzzy system)的结构如图 1-27 所示。

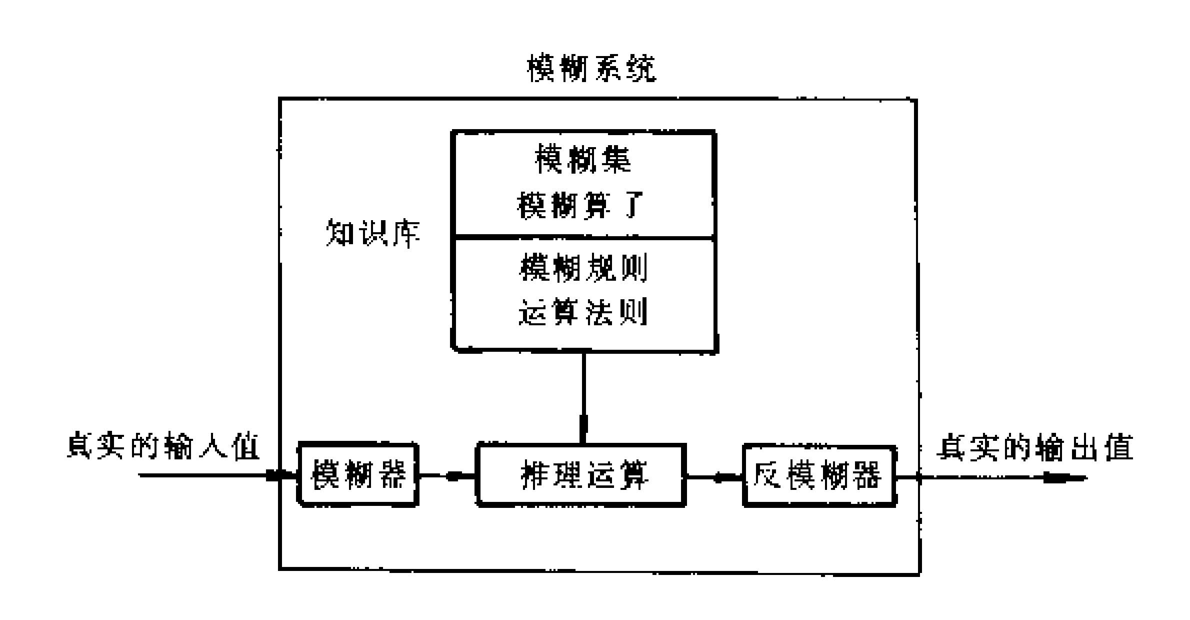


图 1-27 模糊系统结构

在该模糊系统中，包含所有的应用模糊算法和解决所有相关的模糊性的必要成分。它由如下 4 个基本要素组成。

(1)知识库(knowledge base)：它包括模糊集和模糊算子的定义。

(2)推理机制(inference engine)：它执行所有的输出计算。

(3)模糊器(fuzzifier)：它将真实的输入值表示为一个模糊集。

(4)反模糊器(defuzzifier)：它将输出模糊集转化为真实的输出值。

知识库中包含了每一个模糊集的定义，并保持一套算子以实现基本的逻辑(AND，OR，等等)，同时用一个规则信度矩阵表示模糊规则映射。推理单元与模糊器和反模糊器一起，从真实的输入值计算出真实的输出值。模糊器将输入表示为一个模糊集，使得推理单元在存储于知识库中的规则下与之匹配。然后推理单元计算每一规则的作用强度，并输出一个模糊分布

(所有模糊输出集的并),该模糊分布表示真实输出的模糊估计。最后,这些信息被反模糊化(压缩)为单值,该值即为模糊系统的输出。

这些系统非常复杂,它可以被用作一个基本的装置模型、一个控制器或一个估计器,或表示一个性能函数,或用作一个期望的轨迹发生器。它们可以实现一个通用的非线性映射,并由此根据系统输入输出的选择,被用作实现诸多的逼近和分类任务。

模糊系统在其初始化、确认及解释过程中都使用模糊逻辑。一个专业人员可以用一套称为模糊运算法则的模糊产品规则来初始化一个模糊系统。相似地,一个训练后的模糊系统可以用一套模糊算法来解释其行为。当一个模糊系统在解释过程中应用模糊逻辑时,所有与模糊表达式相联系的内在的不精确性都将被完全解决,同时也决定了系统的输入输出行为。本节将探讨怎样实现一个模糊系统,并说明当模糊逻辑使用算子(和/乘)及使用B样条或高斯基函数表示模糊集而且上面涉及的运算都被简化考虑时,模糊系统与一个用B样条或高斯径向基函数(Gaussian Radial Basis Function, GRBF)的神经一模糊网络具有等价性。所有的神经一模糊网络可以简单地作为一个非线性数字处理装置来运作,但它有一个优点,就是可以用一套模糊语言规则来进行初始化和解释。

1.5.2 模糊控制器的设计

模糊控制器(或称模糊推理系统)是直接实现模糊推理算法的专用设备。可以采用软件和硬件两种方式完成一个模糊控制器的功能。当计算量比较小时,可以用软件实现模糊控制器。但对于一些计算量大、实时要求高的控制系统,需要用硬件设备直接实现模糊推理,以达到计算迅速、使用简便的目的。

与模糊推理过程相对应,模糊控制器的设计主要涉及以下几个内容与步骤。

(1)模糊化

模糊化与反模糊化过程可被视为模糊规则与真实世界之间的接口。一个实值输入必须表示为模糊集的形式,才能进行推理计算。而模糊输出集的信息必须被转换为一个单值,这就是模糊推理系统的输出实值。

用一个模糊集表示实值信号的过程称为模糊化。在一个模糊系统处理实值输入时,这个过程是必须的。实现一个模糊器有多种不同的方法,但通常使用最多的是单值化(singleton),它将输入ξ^s转化为一个二值的或具有如下隶属度的确切的单变量模糊集A

$$\mu_A(\xi) = \begin{cases} 1 & \xi = \xi^s \\ 0 & \text{其它} \end{cases}$$

当输入被噪声污染时,模糊集或隶属函数的形状反映了与测量过程相关的不确定性。例如,当峰值与某些测量点的均值相一致,而基宽是标准偏差的函数时,就可以使用三角模糊集。当模型的输入是一个语言表达式时,则必须找到一个可以对等地表示这些语句的模糊集。除

非输入是一个语言表达式时，否则没有理由用与表示如语句"x is small"相同的隶属函数来进行输入的模糊化。后者的隶属函数将被选择以表示模糊语句表述，而输入模糊集反映了同不精确测量过程的不确定联系，一般这两个量是不同的。一个模糊输入分布实际上与一个低通滤波器或一个邻域均值输出等效。当输入集的宽度增加(不精确测量增加)时，则相应增大了邻域输出值的强度，而系统的优点则变得更保守。

(2)建立模糊推理规则

模糊规则表示为"if... then..."条件语句。更清楚的表示是，对多个变化条件的前提经推理产生一个决策结果。在应用中，通常将采用的模糊规则用模糊控制规则表的形式表示出来，在下表中给出了一个模糊控制规则表的例子。两个输入 E 和 EC 各有 7 个模糊语言变量，由此生成 49 条模糊规则。为方便起见，一般按自左到右、自上而下的顺序编号解释其规则，见表 1-1。

表 1-1　模糊控制规则表例

E \ EC (U)	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PB	PB	PM	ZO	ZO
NM	PB	PB	PB	PB	PM	ZO	ZO
NS	PM	PM	PM	PM	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NM	NM	NM	NM
PM	ZO	ZO	NM	NB	NB	NB	NB
PB	ZO	ZO	NM	NB	NB	NB	NB

rule1：if E is NB and EC is NB then U is PB

rule12：if E is NM and EC is PS then U is PM

rule37：if E is PM and EC is NM then U is ZO

其中几个常用的模糊语言变量的符号表示如下：

NB(Negative Big)：负大

NM(Negative Medium)：负中

NS(Negative Small)：负小

ZO(Almost Zero)：几为零

PS(Positive Small)：正小

PM(Positive Medium)：正中

PB(Positive Big)：正大

模糊规则可以通过相关领域的专家给出，也可以通过大量的试验数据给出。无论应用哪种方法，得到的模糊规则都是近似的，因而还需要解决这些规则的协调问题。既要保证模糊规则的完备性，即对于任何模糊输入状态，都必须产生一个模糊控制器的输出，又要保证模糊规则的相容性问题，即模糊规则之间不能得到相互矛盾的结论。解决这两个问题，往往需要一定的工程经验积累和试验数据。

(3)确定权与规则信度

明确地建立模糊规则的权和知识库中模糊规则信度之间的关系是相当重要的。除了模糊表示的隐含关系常常没有被完全考虑之外，有几位研究者已经对上面的描述提出非常相似的结论。例如，考虑规则置信矩阵是二值的，即，对每一个输入集，只有一个规则的信度是非零的，且其值为 1。这个限制意味着，权值只能是有限个值(模糊输出集的"中心")中的一个。对每个模糊输入隶属函数，允许超过一个规则的可信度为有效(active)，应用位于单位区间内的信度，允许相应的权去估计任何位于模糊输出集支集中的值。

此外，当模糊输出集由 $k(\geqslant 2)$阶对称 B 样条基函数定义时，得到如下的关系：

$$\theta_i = \sum_{j=1}^{q} c_{ij} y_j^c = c_i^T y^c$$

其中 y_j^c 是第 j 个输出集的中心，存在 q 个模糊输出集，且规则信度由下式定义：

$$c_{ij} = \mu_{B^j}(\theta_i)$$

即：在权值和相应的模糊规则置信向量之间存在一个可逆(invertible)映射。知识能以两者中的任何一种形式表示，而且在转换时没有信息丢失。权值与 2 阶 B 样条规则置信向量之间的可逆映射如图 1－28 所示。

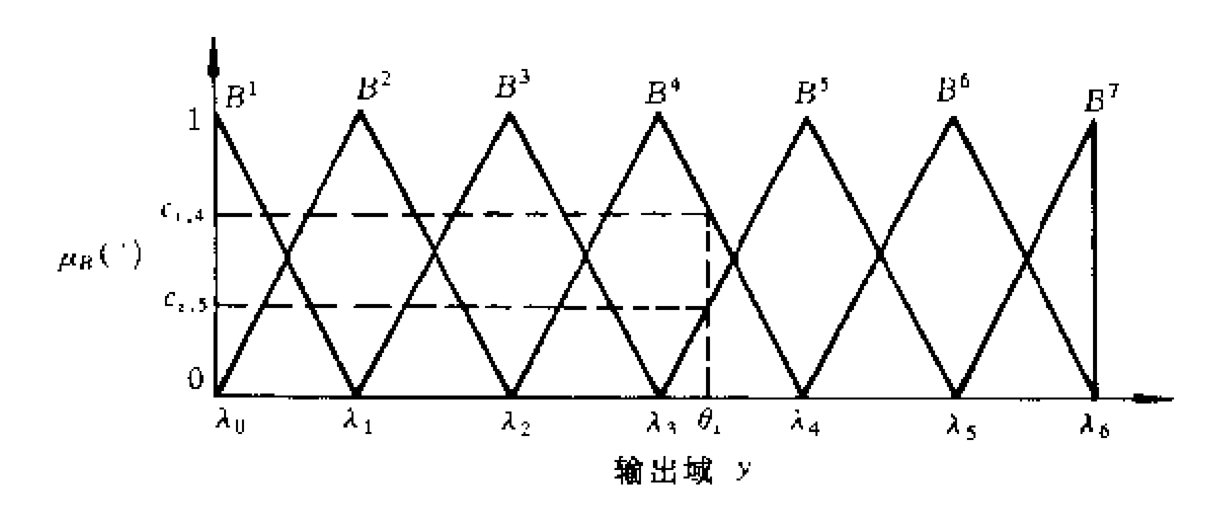


图 1－28　权与相应规则信度向量的关系

这个过程可以被视为对权的模糊化和对单个规则置信向量的反模糊化。而且当使用 $k(\geqslant 2)$阶对称 B 样条函数时，在这个过程中没有信息丢失。

(4)选择适当的关系生成方法和推理合成算法

设计模糊控制器需要选择适当的关系生成方法和推理合成算法。最常用的比较简便的方法是 Mamdani 方法。模糊推理算法与模糊规则直接相关。它的复杂性依赖于模糊规则语句中的模糊集的隶属函数的确定。选择一些简单的又能反映模糊推理结果的隶属函数可以大大简化模糊推理的计算过程。通常高斯隶属函数、梯形隶属函数和三角形隶属函数是使用最多的隶属函数。

(5)反模糊化

当推理过程的输出构成一个模糊输出集 $\mu_B(y)$ 时,就有必要压缩其分布以产生一个表达模糊系统输出的单值。这个过程称为反模糊化,其实现有多种方法。在 1.4.2 中简单介绍过反模糊化的方法,其中两个最常用的方法是最大值平均和重力中心算法,在此对这两种方法作进一步介绍。这两种方法又可分别被归为截断法和代数法两类。因为前者的输出值是在 $\mu_B(y)$ 上具有最大隶属度的值,它是基于对一段信息(或至多几段信息的均值)进行输出估计,故称为截断法。后者使用输出分布中每一点上的标准权值分布进行输出估计,故称为代数法。由于当输入变化时,规则之间有更渐变的转移,因此,重力中心反模糊化算法易于产生一个更平滑的输出曲面。

重力中心反模糊化算法过程的定义如下:

$$y(\xi)=\frac{\int_Y \mu_B(y)y\mathrm{d}y}{\int_Y \mu_B(y)\mathrm{d}y}$$

假定输入由一个单独的模糊集表示,并使用代数截断算子和 B 样条隶属函数时,就能观察到模糊系统怎样进行信息处理这一重要的过程。表示模糊并的加法算子意味着每个模糊集能独立地被反模糊化,使用乘法蕴涵算子则允许模糊规则中的输入输出项能被独立地分析。同样,同时使用标准化规则信度向量、B 样条隶属函数和乘法交算子降低了对如下多变量模糊输入集的简单线性组合的模糊输出计算量

$$y(\xi)=\sum_{i=1}^{h}\mu_{A^i}(\xi)\theta_i$$

因此,模糊输出曲面的形状由模糊输入集的形式决定。每一个权代表一个对特定模糊输入集合的输出估计,对包含于相应规则信度向量中的信息进行反模糊化,即可给出结果。对这些模糊系统的建模和归纳,依赖于所使用的模糊输入集的形式和从输入表示中完全解耦。

1.5.3 神经-模糊系统

在本章以后的内容中,我们介绍神经-模糊系统和自适应模糊系统。这是一类最有发展前景的模糊系统。但对于大多数的读者而言,可以跳过这些内容而几乎完全不影响对模糊控制

的学习和在 MATLAB 中设计和调试模糊控制系统。或者说，在熟悉了模糊控制系统的设计和调试以后，再来阅读本章以后的内容，会更好一些。

神经网络有许多类，但一类重要的神经网络系统能够用模糊规则来初始化和解释其行为。这种神经网络系统称为神经-模糊(neurofuzzy)系统，因为它们在一个单独的、非线性信息处理装置中结合了神经网络(结构和学习算法)和模糊逻辑(模糊性、信息语句与表达)的特性，如图 1-29 所示。

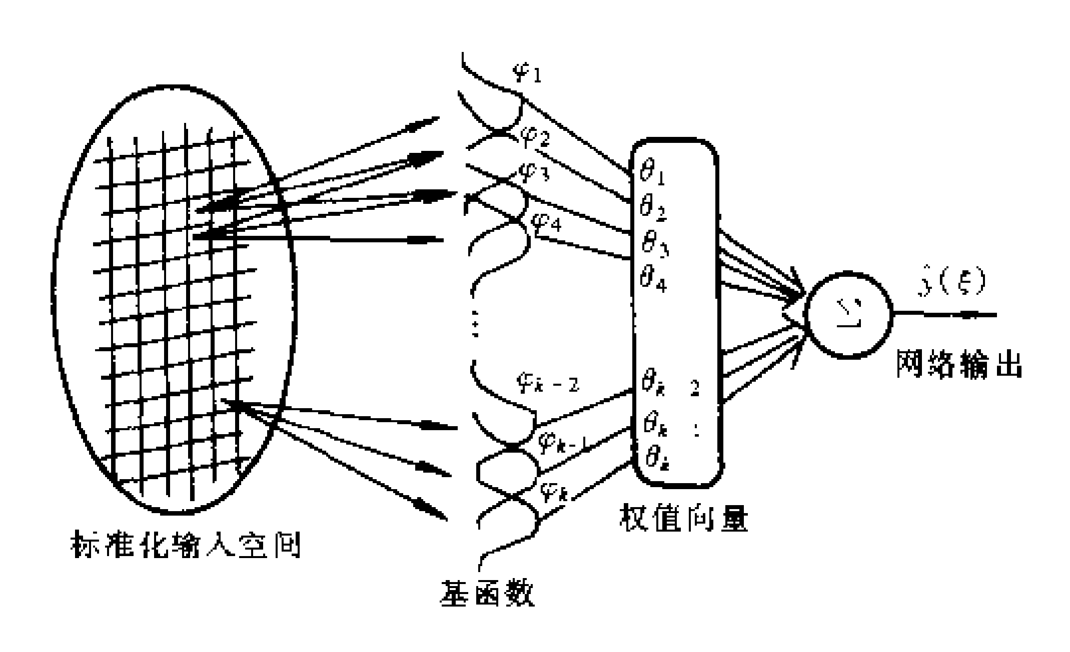


图 1-29 关联记忆网络

图中的分段多项式 B 样条神经网络和高斯径向基函数神经网络一样可以被归类为神经-模糊系统。另一个重要的神经-模糊系统称为混合专家模型(Mixture of Experts Model, MEM)，其中局部最优化函数的逼近是与其它使用标准化加权过程的局部映射相结合的。在此，权是依赖于状态的，并表示局部专家为正确的可能性。

1. B 样条神经-模糊系统

当模糊隶属函数用 k 阶单变量 B 样条定义时，采用了重心反模糊化算法，并使用了代数模糊推理算子，而输入被表示为一个单独的模糊集，此时模糊系统的输出如下式所示：

$$y(\xi) = \sum_{i=1}^{h} \mu_{A^i}(\xi)\theta_i$$

在此式中，多变量模糊集 $\mu_{A^i}(\xi) \equiv \phi_i(\xi)$ 正是定义在 n 维网格(一个完整的规则基)上的多变量 B 样条基函数，而权向量 θ 是可调参数的线性集。因此，模糊系统能被直接映射到图 1-29所示的结构上。其建模和归纳能力依赖于 B 样条基函数的形式和分布。

这种理解对实现模糊系统和其性能的理论分析都有许多重要的结果。说这些神经一模糊系统是泛逼近(universal approximators)，只是一种粗浅的理解，因为一个模糊系统产生分段多项式映射，由 Stone Weierstrass 理论，这个多项式函数集合为泛逼近。同时有必要注意到，无论 B 样条模糊集合个数的增长，还是样条阶数的增长，都可能证实这个结论。与在后面的

"自适应模糊模型"一节中描述的一样，由于可调参数(权值)的集合是线性的，且存在许多标准的学习理论可以用于分析这些自适应系统的行为，这个解释也是相当重要的。它还意味着，当用上式来实现系统，而不是按执行完全的模糊化-推理-反模糊化的过程计算时，系统的计算量会大大地减少。最后，也是最重要的，这种解释允许开发更多的高级神经-模糊学习算法，它能利用训练数据中的冗余数据，并能用于高维建模和控制问题。在"自适应模糊模型"的"离线网络构造"中将对此讨论。

2. 高斯径向基函数神经-模糊系统

当如前面所述实现一个模糊系统时，除了用以单变量高斯函数表示的隶属函数外，模糊神经网络在结构上与标准化高斯径向基函数算法完全等价，其中多变量模糊输入集形成一个单位分解，如下式所示：

$$\phi_j^*(z) = \frac{\phi_j(z)}{\sum_{i=1}^{n}\phi_i(z)}$$

这个修正过的 GRBF 网络有一个标准化因子，这个标准化因子由初始重心反模糊化算法中的分母项引入，它保证了输出曲面质量不被网络形成的内部表示的强度变化所影响。由于最大反模糊化过程的均值仅被分布的相对高度所影响，而不被实际的隶属值所影响，因而它同时使模糊系统的输出标准化。

将多变量模糊输入隶属函数解释为标准化高斯映射有一个重要的优点，即它能被用来克服基函数中心必须定义在 n 维网格上这一局限。可以选择不同的监控和非监控学习规则来选择、优化和聚类模糊集中心，因此当训练数据与测试数据被约束在输入空间中的某一部分时，这种解释是极为有用的。同时，高斯函数是无限可微的，因此可以估计出模糊逼近及其任意阶导数(模型是否已经被足够精确地训练是另一个问题)。另一个重要的属性归因于高斯基函数的局部(但非严格紧)支撑。当只有一个基函数对输出有显著影响时，对输出的标准化能使其支撑明显地具有全局性。但当在邻近定义其它基函数时，则高斯基函数几乎是紧支集的。因此，高斯基函数具有非常灵活的基本形式。

高斯映射是唯一能被写成单变量高斯函数乘积的径向基函数(RBF)，因此，没有其它能被表示为一个神经-模糊系统的 RBF 神经网络。

3. 混合专家神经-模糊系统

在混合专家模型神经-模糊系统中，由于不是用表示真实输出估计的权与每一个模糊输入集相联系，而是用一个存储的 n 维函数与每一个模糊输入集相联系，因此它具有自身的特性。这些存储的 n 维函数表示一个局部模型集或专家，而模糊输入集用于衡量每一个专家对输出的影响，反映模型为正确的可能性。因此，MEM 神经一模糊系统的输出可如下表示：

$$y(\xi) = \sum_{i=1}^{n}\mu_{A^i}(\xi)\theta_i(\xi)$$

其中，$\theta_i(\xi)$在此是一个函数而非一个单值。一般地，每一个$\theta_i(\xi)$是一个线性映射

$$\theta_i = \theta_{i,0} + \theta_{i,1}\xi_1 + \cdots + \theta_{i,n}\xi_n$$

模糊输入集可用B样条或高斯基函数表示，或可以将模糊输入集作为一个输入空间上的的多层 sigmoidal 分解。存储线性函数而不是存储单个的权值，意味着逼近许多光滑函数都要简单得多，虽然这要求n倍的内存。例如，当用一个GRBF神经-模糊系统去逼近一个局部线性函数时，为使逼近误差很小，就要求模糊集的宽度相当大。这反过来意味着，对权值的优化计算是相当病态的。但是，当局部专家模型是线性的且模糊输入集形成一个单位分解时，MEM神经网络模糊系统对这种映射建模是很简单的。

1.5.4　自适应模糊模型

对本章中描述的模糊系统有多种自适应方法。但也有这样一种观点，即当代数模糊算子与反模糊化区域中心和单点模糊化策略相结合时，可以提出一个完整的理论框架，并且这个理论框架能预测这些自适应神经-模糊系统的性能。通常假定要训练的最优参数是那些在参数和系统输出之间有明显和直接(透明的)关系的参数。这个假定不仅改善了学习过程的条件，同时由于更容易理解对自适应神经-模糊系统的修正，因此也简化了验证程序。

神经-模糊系统的三层结构如图1-29所示。

这种三层结构的描述如下式：

$$y(\xi) = \sum_{i=1}^{n} \mu_{A^i}(\xi)\theta_i$$

它说明了参数是怎样被分解到两个不同的范畴，而正是由此决定了输出是线性的(权值)还是非线性的(隶属函数)。调整线性参数集合比训练非线性参数集合要容易得多，而且还存在多种可以证明收敛性并估计收敛速度的学习算法。因此，权(规则信度)是最易于调整的参数。当它们表示对每一模糊输入集的输出估计时，它们是完全透明的，因此当对一个神经-模糊系统进行在线迭代训练时，权值向量是最易于调整的参数集合。为了适应模糊输入集的形状以及在引入和去除隶属函数中可能引发的结构改变，必须使用一个惟一的、可实现的、只能离线进行的复杂非线性训练过程。下面介绍几种神经-模糊系统的训练规则，包括在线和离线的，并讨论其性能。

1. 在线规则自适应

迭代学习程序能用于调整神经-模糊系统的线性参数(权)向量。当数据被接收时，它提供系统期望行为的即时范例，权向量能被反复调整以提高系统性能。最小二乘法(LMS)和标准最小二乘法(NLMS)学习算法是以在瞬时均方误差性能曲面上执行梯度下降为基础的。一般地，这个性能度量由下式给出：

$$J(k) = \varepsilon^2(k) = (\hat{y}(k) - y(k))^2$$

其中 $\varepsilon(k)$ 为输出误差。而且上式产生如下形式的学习规则：

$$\text{LMS}: \Delta\theta(k) = \delta\varepsilon(k)\phi(k)$$

$$\text{NLMS}: \Delta\theta(k) = \delta\varepsilon(k)\frac{\phi(k)}{\|\phi(k)\|_2^2}$$

其中 $\phi(k)$ 是由模糊输入集的输出集所组成的向量，即：$\phi_i(k)=\mu_N(\xi(k))$，δ 为学习速度，$\Delta\theta(k)=\theta(k+1)-\theta(k)$ 是在时刻 k 处的权向量变化。

这些训练程序刷新权向量，使其与归一化输入向量 $\phi(k)/\|\phi(k)\|^2$ 并行，其步长为

$$\text{LMS}: \delta\varepsilon(k)\|\phi(k)\|^2$$

$$\text{NLMS}: \delta\frac{\varepsilon(k)}{\|\phi(k)\|^2}$$

因此，两种学习算法沿着相同的搜索方向刷新其权向量，在沿此路线时只有距离是不同的。两者基于同一个训练数据计算搜索方向及步长。但是，要估计输出误差的哪一部分应归因于权向量的误差、测量误差或建模误差是不可能的。相似地，当两个输入对于输入网格相近或相关，则逐次的搜索方向是高度相关的。因此，虽然训练规则相当简单，但当存在明显的建模和测量误差，并且逐次的训练样本相似时，其性能是退化的。

这种缺陷可以被部分地克服，其方法是采用一个高阶的学习算法集合，用多于一个训练数据来估计当前系统的性能。基于此，可以在搜索方向中引入正交性，并可部分地滤掉测量误差或建模误差。当期望的映射为静态时，有大量的优化技术可以解决此类问题。但是，使用在线学习的主要原因是为了处理时变装置。同时，一般而言，在实时处理中这些标准算法太过繁琐而难于实现。因此，值得考虑怎样导出一套计算花费少的线性最优算法用于时变映射建模。

要获得高阶学习算法，中心问题是其构造一个典型训练数据存储的能力。存储空间的大小是固定的，而且在存储中应当包含最近的（为了对时变装置建模）和最有效的（部分地解除相关性以增加参数收敛的速度）数据。对于数据更新，存在多种不同的方法，但最简单的就是将当前训练数据与所有存储数据相匹配，并清除最老的数据。在此时间和空间的匹配也是加权的概念。只要给出了额外的信息，就能开发许多将经典最小二乘法推广的学习规则。在实践中发现工作得最好的规则，是设置搜索方向与存储中的当前含有最大残差的输入向量相平行，并选择步长以使对所有的数据当前 MSE 为最小。这种方法将搜索方向正交化，并抑制了沿此搜索方向的步数。

2. 调节规则信度

本节中讨论可以怎样有效地训练神经-模糊系统中的权值，而忽略模糊规则信度的表示。由于在权和模糊规则信度之间存在一个学习等价（learning equivalence）关系，因此这是可以调整的，如“权与规则信度”中所示。因此，可以在权空间中执行并训练一个神经-模糊系统，并由于在权和模糊规则信度之间存在可逆映射，故可用模糊规则解释已知函数。但是，直接训练

模糊规则信度也是可能的。在本节中将讨论这是否需要并讨论怎样实现。

神经-模糊系统的输出线性地依赖规则信度矩阵：

$$y(\xi) = \sum_{i=1}^{h} \mu_{A^i}(\xi)\Big(\sum_{j=1}^{q} c_{ij} y_j^c\Big)$$

它可以重写如下：

$$y(\xi) = \boldsymbol{\phi}^{\mathrm{T}}(\xi)\boldsymbol{C}y^c$$

其中$\boldsymbol{C}$为$h \times q$规则信度矩阵。模糊输入集和输出集的中心都是静态的，因此网络输出是在如下定义的空间上的规则信度的线性函数，即

$$\boldsymbol{A} \otimes y^c$$

其中$\boldsymbol{A}$为矩阵，其第k行由$\phi^{\mathrm{T}}(\xi(k))$定义。由于输出集中心不依赖于网络输入，故这个空间是一个当然的奇异空间，因此存在无限个产生最优系统输出的最优模糊规则信度矩阵。甚至当规则信度被约束在单位区间上，且每个规则信度向量是标准的(归一的)，提出的最优化问题仍然不是唯一的。因此，斜率下降过程收敛到一个结果，但它无须一个切实的语言解释。

要取代LMS和NLMS学习规则的期望输出，一个可能的方法是使用期望规则信度向量(desired rule confidence vector)，它通过用与在"权与规则信度"中描述的相似的方法估计期望输出的输出集的隶属度而得到。因此，从期望规则信度中减去当前(加权平均)的规则信度估计，并在如下的模糊LMS或NLMS规则信度更新算法中使用该量，就可构成一个规则信度向量误差。

$$\Delta \boldsymbol{c}_i(k) = \delta\Big(\hat{\boldsymbol{c}}(k) - \sum_j \mu_{A^j}(\xi(k))\boldsymbol{c}_j(k)\Big)\mu_{A^i}(\xi(k))$$

其中$\hat{\boldsymbol{c}}(k)$是期望规则信度向量，它可由模糊输出集的期望输出的隶属度形成。一个用这种学习规则训练的自适应神经－模糊系统与一个类似的基于权的、其权向量是用LMS算法调整的网络是学习等价的。

总的来说，可以为神经-模糊系统找到等价的模糊规则信度训练算法，但其实现的代价更大，而且由于最优化问题内在的奇异性，它们的透明性要小些。

3. 离线网络构造

确定一个神经-模糊系统的结构是一个十分复杂的非线性迭代优化问题，对此必须找到哪些输入更重要、是否一个单独的网络能够用两个或更多个更简单的网络来代替，以及各(子)网所构成的内在表示(模糊集的个数和形状)是什么。在任何建模工作中，这都是一个极为重要的任务，因为如果网络结构不适当，网络要么是过度参数化而含噪建模，要么是缺乏灵活性而不足以存储要求的信息。另外，构造一个适当的内在表示可以将下面4个期望的性质赋予合成系统：① 提高系统的归纳能力；② 减少所需的训练数据个数；③ 优化网络条件；④ 简化网络结构以得到更透明的规则库。

在本节中将讨论怎样利用附加的和局部的冗余数据以解决此问题。

在前面的描述中，这个通常神经-模糊网络都有的在中高维输入空间中定义的逼近函数的问题，不会立即显示出来。但当一个复杂的实际应用系统被开发出来时，这就是一个严肃的问题。例如，考虑用定义在每个（输入输出）轴上的 7 个单变量模糊集合对一个 5 维映射建模，则神经-模糊系统所需要的数据源为 $7^6 \approx 120\ 000$ 个规则信度或 $7^5 \approx 17\ 000$ 个权值。在每一种情况，系统至少需要 20 000 个分布较好的训练数据，以在整个输入空间和许多实际情况下正确地推广，要获得哪怕只是一小部分这些信息也是不可能的。这个问题通常称为维数灾难（curse of dimensionality）。

4. 全局冗余

要减轻在神经-模糊系统中发生的维数灾难，最简单的方法之一，是尝试利用在期望函数中产生的任何全局冗余。要达到这个目的，最简单的方法是只引用那些在模型中贡献了大量信息的输入。用忽略冗余输入来减小输入空间大小的方法可以大大简化神经-模糊系统结构。另一个简化系统结构的方法是利用任何的加性冗余。在最简单的情况下，网络可以由如下形式的非线性单变量模型的线性组合构成：

$$y(\xi) = \sum_{i=1}^{n} s_i(\xi_i)$$

其中 $s_i(\cdot)$ 为单变量非线性神经-模糊子网络。整个网络极为简单，其计算量以 n 倍线性增长。但它只能对不同输入变量之间的加性交互作用建模，即要求没有 $\xi_1\xi_2$ 形式的交叉乘积多项式。更一般的模型形式如下：

$$y(\xi) = \sum_{i=1}^{r} s_i(\xi_i)$$

其中 r 是（可能是多变量）神经-模糊子网络的个数，其输入向量 ξ_i 属于一个维数更小的子空间。这种方法试图用一个维数更小的神经－模糊子网络的线性组合来逼近基本期望函数。

当一种迭代方式是用一个非常简单的模型开始，且在存储中有一个单变量网络集合，则在迭代中可以辨识出加性相关性。存储中的每个单变量网络都可被视为神经-模糊系统中可能的单变量子网络，而减少当前输出误差最多（依据其复杂性定义权值）的单变量子网络也包含在神经－模糊系统中。要注意，这些网络能用标准的优化技术（例如共轭梯度法）进行训练。由于加性冗余被明确建模，因此改善了优化问题的条件。这个程序迭代构成由单变量非线性子网络线性组合而成的神经-模糊系统。如果允许在一个当前子网络和另一个存储中的单变量网络之间、或在一个当前子网络和另一个包含在当前模型中的子网络之间存在一个张量乘积运算，则可以发现维数更高的子网络。这种情况将初始子网络增加了一维，并意味着新的子网络能精确地存储前两个子网络中的信息，同样，它还能对新的交叉乘积项建模。在实践中，在任何时刻都有可能或者包含一个新的单变量网络，或者形成一个张量乘积运算。而且，最具

有统计意义(statistically significant)的子网络也包含在当前的模型中。

5. 局部划分

在开发一个自动网格构造算法时,有必要设计一个能够优化模糊隶属函数的个数与位置的程序。前面描述的归纳学习程序决定了哪些输入变量在精确预测装置性能当中更为重要,但一般而言,还有必要改进(增加、修正或简化)子网格的结构。然而,开发一个设计优良的迭代优化程序依赖于每个子网络使用的表达式。可作的最简单的假定是多变量模糊集被定义在网格上(用每个单变量模糊子网络的张量乘积)。在此假定下,每个单变量模糊子网络能被分别优化,而且从它们的张量乘积中生成一个新的多变量子网络。当使用B样条模糊隶属函数时,可以在节点向量中加入一个新的节点(例如在前两个值的中间),而且这样做将引入一个新的基函数,并局部修正前面的 k 个隶属函数。如图1-30所示。

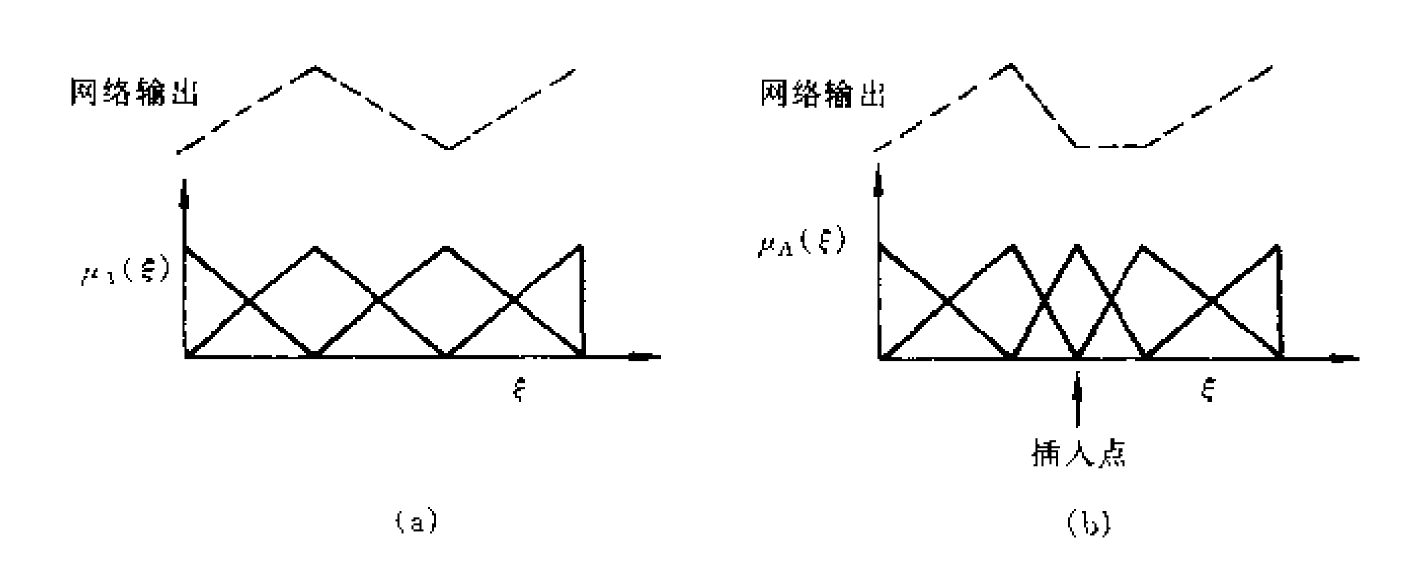


图1-30 在网格中插入一个新的B样条隶属函数(节点)
(a) 插入前的网络输出;(b) 插入后的网络输出

在这种情况中,一个新的B样条模糊隶属函数(节点)被插入一个单变量子网络中。这个过程局部修正了现存的模糊隶属函数,并在模型中增加一个额外的自由度(另一个分段线性段)。在单变量基函数中加入一个单节点,与在一个2维网格中增加一条新的线段或在一个 n 维输入空间中增加一个 $n-1$ 维超平面是等价的。如果约束新节点的值,令其发生在相邻值中间,并估计可能的位置,则有可能决定哪一个优化可以对未建模的数据产生最优拟合,以修正子网络并重新训练整个网络。

但有时候,当上一步归纳学习程序并不总是最优的并可能出错时,子网络就可能变得太过灵活了。在模型中包含了不应包含的输入,或节点插入程序仅从区间中值集合中选择一个可能值,则可能形成过于复杂的子网络。因此,在限制网络复杂性中,网络去除是一个很重要的工具。可以删除节点(模糊隶属函数),并将子网络分解为维数更小的子网络,只要这些子网络中的一个可以产生一个与更复杂的模型一样适于数据的模型,则可以简化整个神经-模糊系

统。

最后，有必要考虑前面设计中的局限性。大多数神经-模糊系统都有一个定义在网格上的基函数集。虽然通过对所有全局加性冗余直接建模，上面的程序可以部分地克服这个局限，但每一个子网络仍然建立在一个维数更小的网格上。应用更复杂的输入空间分割技术可以进一步部分地克服这个局限，虽然其中可以被解释为模糊规则的大部分仍然依赖于将输入空间在平行于轴的方向上分割，如图 1-31 所示。即使允许了这些可以利用局部冗余的分割策略，也仍有必要在将输入表示到每一个子网络之前，将输入旋转并解除其相关性。当考虑了这些可能的局限时，基本的网络结构算法才可能为其它高维模糊系统产生简洁的中等尺度模型。

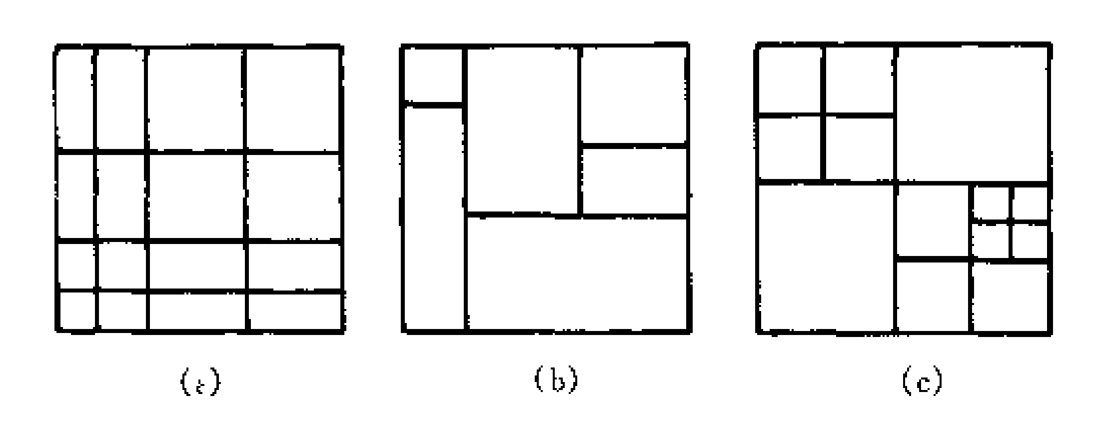


图 1-31　输入空间的网格划分、树状划分和层级划分

(a) 网格划分；(b) 树状划分；(c) 层级划分

1.5.5　自适应模糊控制系统

到目前为止，我们介绍了在一个模糊系统中实现模糊算法的基本理论，并讨论了应用不同的隶属函数、算子等的意义。但是，当这些学习模糊系统被用于自适应控制时，它们必须被用作全局设计计划中的一个部分。下面我们了解一下几种不同的直接和间接自适应模糊控制器，并提供几种不同的实现方法。首先从考虑装置建模的基本问题开始。

1. 自适应模糊建模

相对而言，将一个模糊系统用作一个自适应装置的模型是比较简单的，因为可以直接利用期望的训练信号(测量得到的装置输出)，且只有模糊系统的结构需要被决定。从而学习过程是一个线性优化过程，且当数据满足边值约束时，可以证明学习过程将收敛至全局最小。在图 1-32(a)中给出了最简单的结构。其中模糊装置模型的输入由控制输入的时延测量值和装置的输出构成。在此情况下，假定训练数据由如下描述的过程产生：

$$y(k) = f(\xi(k))$$

其中 $f(\cdot)$ 是一个未知的非线性映射，它表征了未知装置。

$\xi(k)=(y(k-1),y(k-2),\ldots,y(k-n_y),u(k-1),\ldots,u(k-n_u))$ 是装置的信息向量。

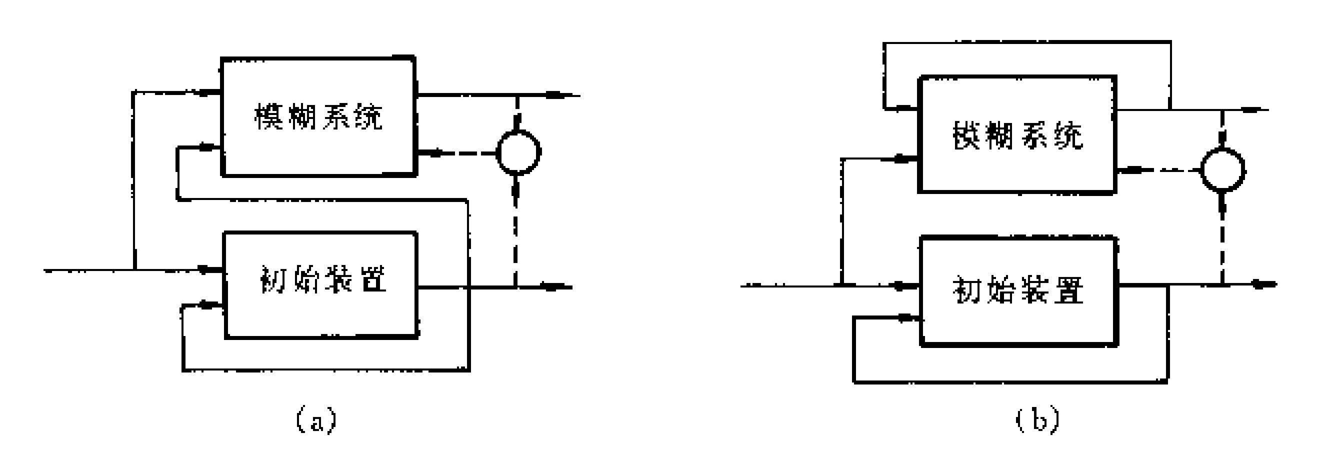


图 1-32　两种基本的模糊装置模型
(a) 采用测量值作为输入的模型；(b) 采用自身输出作为输入的模型

故一个模糊模型可如下构成：

$$\hat{y}(k)=\hat{f}(\xi(k))$$

其中 $\hat{f}(\cdot)$表示模糊系统。输入向量由装置的输出值和控制输入组成。假定测量是正确的，并已得到模型阶数 n_y,n_u 的正确估计，而控制输入为持续激励，则模糊系统将令人满意地逼近 $f(\cdot)$。甚至即使这些条件并不满足，也可以证明模糊系统能跟踪装置的输出（虽然此时无法将网络适当地推广），即

$$\lim_{k\to\infty}\varepsilon(k)=0$$

当测量得到的输出被噪声污染时，建模过程将可能是有偏的。而当输入被污染时，将无法判断输出误差是源于参数不匹配还是源于输入不正确。为克服此困难，模型的输入采用其自身的输出而不用测量值（假定模糊系统的输出足够接近真实值），产生一个如下的模糊模型：

$$\hat{y}(k)=\hat{f}(\xi(k))$$

其中，$\xi(k)=(\hat{y}(k-1),\hat{y}(k-2),\dots,\hat{y}(k-n_y),u(k-1),\dots,u(k-n_u))$是模糊系统的信息向量，如图 1-32(b)所示。在当前的跟踪误差可能被前几步的参数误差所影响而非最近一步的参数误差所影响时，训练这些循环的模糊模型的学习规则是非线性的。

这些模糊系统能用于对一个专家算子建模，这个专家算子知道怎样去控制难于建模的过程，或直接构造这个过程的逆向模型，或更直接地预测这个未知过程的状态，这个未知过程的状态随后被用于预测控制算法或间接控制器的设计。所有这些应用的成功依赖于模糊规则库的结构（对数量 n_y,n_u 和非线性映射构成的估计）和在"离线网络构造"中描述的可用于此的算法。

2. 间接自适应模糊控制

自适应控制算法可以被粗略地分为直接规划和间接规划。在后一种方法中，不用形成一个明确的控制器模型，此时自适应模糊模型被用于间接地构成一个控制信号。在过去的几年中人们已经开发了大量的间接自适应模糊控制系统（此前所有的自适应模糊系统都是直接的）。

3. 仿射系统

人们为了对一类所谓的仿射系统开发控制算法,已经做了大量的工作。在控制信号中,仿射系统是线性的,并可表示如下:

$$y(k) = f(\xi(k)) - g(\xi(k))u(k-1)$$

其中 $f(\cdot)$、$g(\cdot)$是非线性函数,且 $\xi(k) = (y(k-1), y(k-2), \cdots, y(k-n_y), u(k-1), \cdots, u(k-n_u))$。在一个有两个以上的子网络、其中一个子网络依赖于 $u(k-1)$、且 $u(k-1)$由两个2阶B样条建模(即在紧域上是线性的)的模糊系统中,可以很容易地表示这个由初始装置分解为两个子模型、其中一个模型与当前控制信号相乘的加法与乘法分解。因此,前面描述的网络构造算法能被用于精确地决定哪些输入更重要以及它们应被怎样表示。这种模糊装置模型的表示如下:

$$\hat{y}(k) = \hat{f}(\xi(k)) + \hat{g}(\xi(k))u(k-1)$$

假定 f,g 能满足一定的条件,例如平滑及 g 不为0,并且知道模糊子网络能充分地逼近 f 和 g,则有可能在确定等价性原理(certainty equivalence principle)的基础上设计一个控制器。因此,用 $y_m(k)$代表 k 时刻的期望输出,则由下式计算得到控制信号:

$$u(k-1) = \frac{y_m(k) - \hat{f}(\xi(k)) + \boldsymbol{d}^{\mathrm{T}}\varepsilon(k)}{\hat{g}(\xi(k))}$$

其中 $\boldsymbol{d}^{\mathrm{T}}$ 是一个向量,使得多项式 $q^n + d_1q^{n-1} + \cdots + d_n$ 的根都位于一个单位圆内,且 $\boldsymbol{\varepsilon}(k) = (\varepsilon(k-1), \cdots, \varepsilon(k-n))$是一个延时输出误差向量。这样,则有可能为模糊子网络 $\hat{f}$,$\hat{g}$ 设计简单的李亚普诺夫更新规则,并可以由此证明系统的稳定性。严格地说,这还需要一个稳定性监控程序,只要输入位于模糊系统的区域之外,它就发挥其作用。

仿射系统最有吸引力的特征之一,是其结构能用输入输出线性化技术决定,这为输入空间尺度给出了一个上界。

4. 模糊模型转换

我们不将装置模型假定为仿射结构,而是在控制输入中将装置模型简化,并直接用转换模糊规则库、或间接用数字搜索程序、或通过配置已知函数,将模糊装置模型转换。在所有的算法中,都用一个模糊模型映射当前装置状态并控制下一个状态的输入。假定一个参考模型能提供一个期望的行为,即如果 $y(k) = y_m(k)$已知,则必须将规则库转换以找到 $u(k-1)$。其中第一个方法(即利用输入输出都是由模糊集表示、且蕴涵算子是可逆的这一事实,以直接地转换模糊库)看起来很有吸引力,因为控制器是由一个明确的模糊规则集合构成。但是,这种方法的普适性较差,因为规则信度向量不再是标准的(归一的),而且控制系统不能在输入域边界上正确地推断信息。

间接用数字化搜索程序转换模糊规则库是最精确的方法,虽然它可能占用较多的时间,并可能在映射中存在不连续(因为模糊规则信度未被初始化),因此不能使用假定了某种平滑特

性的搜索程序。一个被证明有效的技术是，在均方差意义下对映射 $u(k-1)\rightarrow y(k)$ 配置（全局）线性函数，然后转换此线性关系。但是，这个方法隐含地假定在装置中存在一个仿射结构，而如果这是事实的话，则在上一节中提到的技术可能更适当一些。

5. 其它方法

另外有许多方法可以将模糊模型用作其控制设计程序的一部分，例如 d 步超前预测控制算法，它应用优化算法来寻找一个使预设的性能函数值最小的控制信号序列。这个程序并不明确地使用模糊系统的语言属性，但是与一个高效执行相结合的局部学习程序已足以调整其使用。同样，还有一种方法是用模糊系统来对一个线性系统的（非线性）放大系数建模，并进一步用局部线性模型来完成控制器设计。在此使用了模糊系统，因为它们对可调参数是线性的，而且优化程序的条件良好。可能在初始化这两种系统时要用到对过程的专家知识和模糊知识，而且，虽然这是一个有用的特征，但它适于这些工作的原因是源于网络的学习能力。

6. 直接自适应模糊控制

一个直接自适应模糊控制器可以直接地建立一个模糊控制器。模糊控制器将当前状态和期望状态信息映射到一个可取控制信号。最简单的也是用得最多的模糊控制器是用于测量（从一个设定点处的当前的偏离）误差和误差率，而基于输入信号的模糊系统或者输出控制信号，或者输出在控制中可取的变化，如图1-33所示。

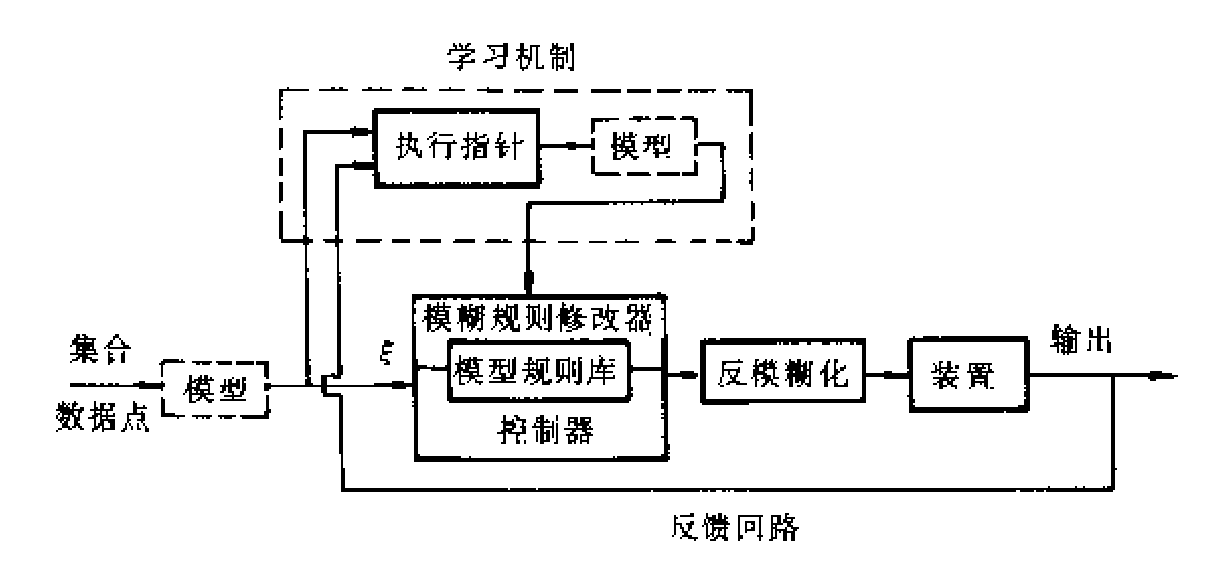


图1-33 直接模糊控制系统

直接模糊控制器实现了一个非线性控制算法，可以使用模糊的语言化规则将这个非线性控制算法以一种很自然的方式进行初始化和有效化。事实上，目前被开发出的模糊控制系统主要是静态的，模糊系统更可取的惟一可能的原因大概是其常识性，也就是模糊规则能以一种自然的方式被实现和迭代。自适应模糊控制器试图通过学习装置如何响应不同的控制动作来获得其规则，并因此更新其控制策略。因此，可以简单地说，直接（和间接）模糊控制器是一种非线性学习系统，而且它们同常规方法一样，服从同样严格的收敛性、稳定性和鲁棒性分析。

许多直接自适应控制器是基于误差的，每一输入信号都是基于当前装置输出值和要求的

设定值(例如其导数和/或积分)之间的误差,因此它就像一个标准的非线性 PID 控制器。但是,使用直接自适应模糊系统的一个主要原因是希望控制非线性装置,除非已经采取特殊的预防措施,比如根据控制点和装置的局部动态行为来改变模糊集的定义,从而,基于误差要求的控制器映射将依赖于状态点。这个程序同与线性控制器增益时序安排是等价的,因为改变模糊集的定义将有效地改变整个系统的放大系数。

7. 学习机制

在建立一个直接自适应控制系统时,必须回答的基本问题是,输出误差中的哪一部分是源于不正确的控制信号?例如,用原始的阶跃响应数据去适应控制器是不可行的,因为它等价于要求装置要获得系统状态中的瞬时过渡。因此,信号必须用一个近似的过程模型(在某种意义上)加以滤波。为使要求的装置输出信号是可实现的,需要将这个过程模型置于控制器之前,否则的话,则需要将这类知识隐含在学习机制当中。与其说这些直接自适应模糊控制器不需要一个过程模型,不如说它要求一些关于动态行为的先验知识(不论是定量的还是定性的)是可得到的,而且整个控制系统的性能将取决于这些特定的品质。

最早的自组织模糊控制器使用一个性能指标和一个装置的逆雅可比估计。性能指标用于将装置的输出测量(例如与设定点的偏差量和当前的误差变化速度)与表征系统性能的信号联系起来。装置的逆雅可比模型将性能测量映射到要求的控制信号变化上,这种要求的变化是改善系统性能所必要的。这个过程如图 1-33 所示,除了第一个装置模型(它转换要求的阶跃响应)未被使用外。实际应用已经证明这个程序工作良好。

显然,在训练直接自适应控制模糊控制器中,使用性能指标和(逆)过程模型不是惟一的方法,虽然在许多该学科的文献中,人们希望保持这个模糊训练机制。相反,我们相信,测试更多的常规的具有收敛性、稳定性和鲁棒性结果的学习算法(有时它们更易于扩展到多变量情况)可以取得很大的进展。例如,如果装置的输出线性地依赖于控制信号,但非线性地依赖于其它的状态和时延控制输入,我们就能开发出与在自适应控制文献中传统使用的相似的直接学习算法。模糊控制器是简单的非线性数字映射,可以使用语言化模糊规则将其初始化并激活。语言化表示的重要性在于它是控制器知识库和设计者之间的交互界面,而自适应模糊控制器的性能(收敛速度)依赖于模糊系统的结构和学习机制,应当选择两者使整个自适应过程尽可能地简单。在这个过程中,是否使用传统的“模糊”概念(min/max 算子、性能指标等)并不重要。

1.6　常用的专业词汇

本节中简要地解释了一些从模糊逻辑中派生出来的专业化术语。

模糊集(fuzzy set):一个包含只具有部分隶属度元素的集合。

模糊单点(fuzzy singleton)：一个模糊集，其隶属函数只在某一特定点上为1，在其余点上均为0。

单点输出函数(singleton output function)：一个输出函数，它由一个单个数字而非一条连续曲线给出。在模糊逻辑工具箱中仅支持它作为一个零阶 Sugeno 模型的一部分。

隶属函数(MF,membership function)：一个函数，它指定一个给定的输入属于一个集合或与一个概念相关的程度。

隶属度(degree of membership)：隶属函数的输出值。该值限制在0和1之间，又称为隶属值或隶属等级。

模糊算子(fuzzy operators)：即 AND，OR 和 NOT 运算符。它们又称为逻辑连接。

T 范式(T-norm)：一个二输入函数，它描述了模糊交(AND)算子的一个超集，包括最小值、代数积和任何若干参数的 T 范式。

T 协范式(T-conorm)：(又称为 S 范式(S-norm))一个二输入函数，它描述了模糊并(OR)算子的一个超集，包括最大值、代数和和任何若干参数的 T 协范式。

前提(antecedent)：模糊规则的前一部分(或 if 部分)。

结论(consequent)：模糊规则的后一部分(或 then 部分)。

激励强度(firing strength)：模糊规则前一部分的满足程度。激励强度可以是 AND 或 OR 操作的输出，它为规则构造输出函数，又称为满足度。

满足度(degree of fulfillment)：参阅激励强度。

模糊化(fuzzification)：用隶属函数对模糊变量产生隶属度的过程。

反模糊化(defuzzification)：将模糊推理系统的输出转化为一个明确的输出的过程。

聚类(aggregation)：在 Mamdani 模糊推理系统中为反模糊化预备的对每个规则结果的组合。

模糊 C 均值聚类(fuzzy C-means clustering)：一种数据聚类方法，其中每一个数据点由一个隶属等级指定它属于一个聚类的程度。

消去聚类法(subtractive clustering)：一种由检测输入输出训练数据中的聚类来自动地产生模型推理系统的技术。

Mamdani 型推理(Mamdani-type inference)：一种模糊推理，从每个规则的结果中得到的模糊集通过聚类运算与结果模糊集相组合后，被反模糊化以产生系统输出。

Sugeno 型推理(Sugeno-type inference)：一种模糊推理。其中每个规则的结果是输入的线性组合，而输出是结果的加权线性组合。

蕴涵(implication)：或称隐含。在 Mamdani 型 FIS 中，在前提基础上，对结果中的模糊集合进行塑造的过程。

模糊推理系统 (FIS,fuzzy inference system)：使用模糊推理将一个输入空间映射到一个

输出空间的系统的总称。

自适应神经-模糊推理系统(ANFIS,adaptive neuro-fuzzy inference system):一种基于训练数据对 Sugeno 型推理系统的自动调节技术。

NB(Negative Big):负大,常用的模糊变量语言值(下同)

NM(Negative Medium):负中

NS(Negative Small):负小

ZO(Almost Zero):几为零

PS(Positive Small):正小

PM(Positive Medium):正中

PB(Positive Big):正大



MATLAB 模糊逻辑工具箱

MATLAB 是 MathWorks 公司于 1982 年推出的一套高性能数值计算和可视化软件。它集数值分析、矩阵运算、信号处理和图形显示于一体，构成了一个方便的、界面友好的用户环境。MATLAB 强大的扩展功能为各个领域的应用提供了基础，由各个领域的专家学者相继开发了 MATLAB 工具箱。模糊逻辑工具箱正是其中的一个。模糊逻辑的创始人 Lotfi A. Zadeh 对模糊逻辑工具箱作出了高度的评价。他说："由于模糊逻辑工具箱为系统结构的计算提供了模糊规则，因此它极大地增强了人们的推理能力。并由于 MathWorks 公司无可匹敌的 MATLAB 软件及其图形界面技术，人们的推理能力更是进一步加强。"同时他说："模糊逻辑工具箱在各方面都给人以非常深刻的印象。它使得模糊逻辑成为对于智能系统的概念和设计的有效工具，而且它易于掌握，使用方便。"这些也正是广大工程技术人员在使用 MATLAB 及其模糊逻辑工具箱以及其它工具箱时所深刻感受到的。

在此，我们给出模糊逻辑工具箱的最新介绍，以方便读者的应用。

2.1 模糊推理过程

在模糊推理的过程中，明确地描述了怎样用模糊逻辑将给定的输入映射到输出。其中涉及了在第 1 章中描述过的所有内容，如隶属函数、模糊逻辑算子和 if-then 规则等。

在模糊逻辑工具箱中，可以实现两种类型的模糊推理系统，即 Mamdani 型模糊推理系统和 Sugeno 型模糊推理系统。如在第 1 章中所述，这两种推理系统在确定输出的方法上略有不同。

在模糊逻辑工具箱中，模糊推理过程由 5 个部分构成，即输入变量的模糊化、前提中模糊算子(AND 或 OR)的应用、从前提到结果的蕴涵关系、模糊规则结果的聚类和反模糊化。

下面的模糊推理图表示了一个二输入一输出三规则模糊系统模糊推理过程的各部分。图 2-1中给出了信号在模糊推理图中的流向。

数据流从左下部的输入开始，然后穿过所有的箭头(或规则)，再下转至右下部完成输出。这条路径非常紧凑地表现了模糊推理过程的所有东西，包括从语言变量的模糊化到聚类输出

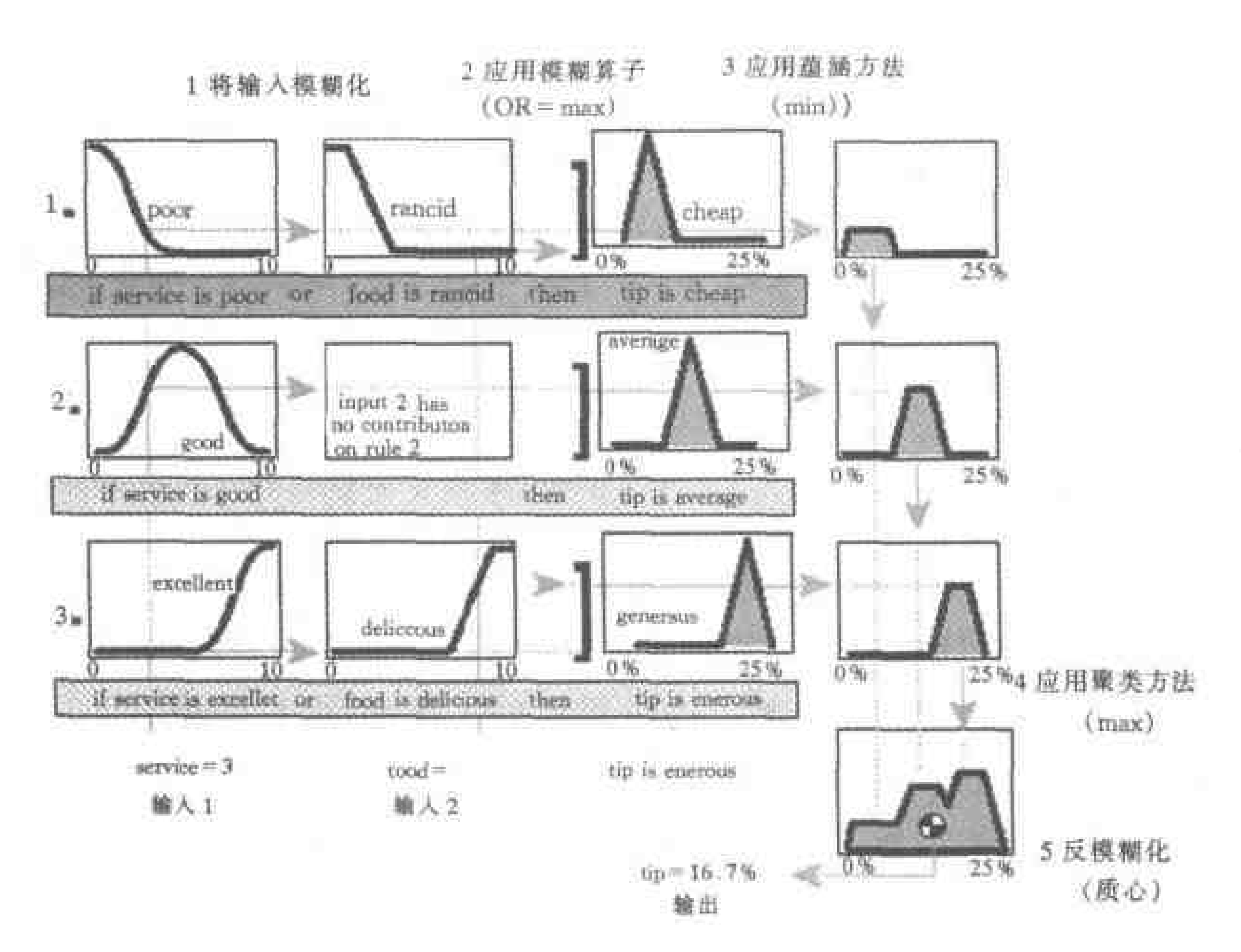


图 2-1　模糊推理过程

的反模糊化。

1. 模糊推理过程的步骤

(1) 第 1 步:输入的模糊化

模糊推理过程的第一步,是获取输入,并确定它们通过隶属函数而属于每个适当的模糊集合的隶属度。在模糊逻辑工具箱中,输入总是一个明确的数值,而这个数值被限制在输入变量的论域(范围)中。输出则是一个限定语义集合中的模糊隶属度(总在 0,1 之间)。事实上,输入模糊化的概念与查表或函数估计的概念是等价的。

一个模糊推理系统的推理基础建立在多个模糊规则之上,每个规则依赖于将输入解析到若干个不同的模糊语义集合之上,如“个子不高”、“个子有点高”、“个子很高”、“电压太低”、“电压太高”等等。必须首先根据各语义集将输入模糊化,才能进行规则的计算,例如到什么程度个子才真正可以算“很高”。对所有的规则,各个输入均按照相应的隶属函数被模糊化。

(2) 第 2 步:应用模糊算子

完成了输入模糊化,我们就知道对于每个模糊规则,前提中每一部分被满足的程度。如果一个给定规则的前提有多个部分,则可用模糊算子来获得一个数值,这个数值表示前提对于该规则的结果。随后,这个数值被应用于输出函数中。模糊算子的输入是两个或更多个从模糊化输入变量得到的隶属值,而输出是一个单独的真实值。

在模糊逻辑中有多种明确定义的方法可以实现模糊交(AND)和模糊或(OR)算子。模糊逻辑工具箱中内置了两个模糊交方法：min(最小值)和 prod(乘积)，同样还内置了两个模糊或方法：max(最大值)和 probor(OR 概率统计方法)。OR 概率统计方法又称为代数和方法，其运算按如下等式进行：

$$\mathrm{probor}(a,b)=a+b-ab$$

除了这些内置方法，用户还可以自定义实现 AND 和 OR 的方法。其方法是建立一个函数，并将其设为自己的选择项。后面对此将有更多介绍。

例如，在一个二输入三规则的实际工作系统中，如果前提中的两个部分分别产生两个不同的模糊隶属度 0.0 和 0.7。模糊 OR 算子只要简单地选择二者中的最大值，就完成了对某一规则的模糊运算。在这种情况下，如果我们使用 OR 概率统计方法(代数和方法)，如上式所示，则其结果仍为 0.7。

(3) 第 3 步：模糊推理

在应用模糊蕴涵进行模糊推理之前，必须注意规则的权。每一规则都有一个权(0 到 1 之间的一个数值)，它被用于前提给定的个数。通常权值为 1，有时用户可能想将某个权值改为一个非 1 的值，以改变相应规则与其它规则之间的相对权值。如果某一规则的权值为 0，则它在模糊推理过程中不产生任何作用。

一旦为每一规则指定了适当的权，就实现了蕴涵。其结果之一，是模糊集由隶属函数表示，隶属函数规定了属于它的语义特征的权值。使用一个与前提(一个单值)相关的函数，则结论被重新整形。模糊蕴涵过程的输入是前提给定的一个单值，输出是一个模糊集合。由此实现了对每个规则的蕴涵过程。对此，工具箱有两个内置的方法，它们与 AND 方法使用同样的函数：min 和 prod，前者截断输出模糊集，后者规定输出模糊集的尺度。

(4) 第 4 步：聚类输出

由于决策是在对模糊推理系统中所有规则进行测试的基础上作出的，故必须以某种方式将规则结合起来以作出决策。聚类就是这样一个过程，它将表示每个规则输出的模糊集结合成一个单独的模糊集。只有在模糊推理过程的第 5 步也就是反模糊化之前，才对每个输出变量进行一次聚类处理。聚类过程的输入是对每个规则的蕴涵过程返回的截断输出函数，其输出是一个输出变量的模糊集合。

由于聚类方法是可交换的，因此在聚类方法中，规则的执行顺序无关紧要。工具箱有 3 个内置方法：max、probor 和 sum。其中，sum 执行的是各规则输出集的简单相加。

(5) 第 5 步：反模糊化

反模糊化过程的输入是一个模糊集，即上一步中的聚类输出模糊集，其输出为一个单值。在各中间步骤中，模糊性有助于规则计算，但一般而言，对各变量最终的期望输出是一个单值。但是，模糊集的聚类中包含了许多输出值，因此必须将其反模糊化，以从集合中解析出一个单

输出值。

反模糊化最常用的方法是重心计算，它计算并返回曲线下区域的中心。工具箱中有 5 个内置方法：重心法(centroid)，二等分法(bisector)，中间最大值法(middle of maximum)(输出集最大值的平均)，最大最大值法(largest of maximum)，最小最大值法(smallest of maximum)。

2. 自定义模糊推理

模糊逻辑工具箱的基本目的之一，是建立一个开放的、易于修改的模糊推理系统结构。因此，模糊逻辑工具箱被设计为在描述过程的基本约束之内，使用户尽可能自由地为自己的应用程序设定模糊推理进程。例如，用户可以用自己的 MATLAB 函数代替在上面详细说明的 5 步中默认使用的任何函数，亦即用户可以定义自己的隶属函数、AND 和 OR 方法、蕴涵、聚类方法及反模糊化方法。在下一节中将介绍用模糊逻辑工具箱提供的工具来建立并实现一个模糊推理系统。

2.2 模糊逻辑工具箱的图形界面工具

在此我们介绍模糊逻辑工具箱提供的图形用户界面(GUI)工具。尽管可以严格地以命令行方式使用模糊逻辑工具箱，但总的来说，使用图形工具建立模糊推理系统更容易些。

模糊逻辑工具箱有 5 个主要的 GUI 工具可以用来建立、编辑和观察模糊推理系统。如图 2-2所示，这 5 个 GUI 工具中包含了 3 个编辑器，即模糊推理系统(fuzzy inference system，FIS)编辑器、隶属函数编辑器、模糊规则编辑器；2 个观察器，即模糊规则观察器和输出曲面观察器。这些图形化工具相互之间是动态连接的，因此，在使用中，只要模糊推理系统任何一个 GUI 的参数或性质被用户修改，其它打开的任何 GUI 中相应的参数或性质都将自动地被改变，这一点极大地方便了用户对自己的模糊推理系统进行调试。

在任何一个给定的系统，都可以使用 5 个 GUI 工具中的几个或全部。

除了这 5 个主要的 GUI，工具箱中还包括 ANFIS(神经-模糊推理系统)图形用户界面编辑器，它用于建立并分析 Sugeno 型自适应神经-模糊推理系统。在后面“神经-模糊推理编辑器 ANFIS”一节中将对 ANFIS 编辑器作详细介绍。

模糊逻辑工具箱 5 个主要 GUI 工具的功能如下。

FIS 编辑器：处理系统的高级问题，如：有多少个输入和输出变量？其名称是什么？等等。模糊逻辑工具箱输入变量的个数没有限制。但是，输入变量的个数会受到计算机内存的限制。如果输入变量的个数太多，或隶属函数的个数太多，则可能难于使用其它 GUI 工具来分析 FIS。

隶属函数编辑器：用于定义与每个变量关联的隶属函数的形状。

模糊规则编辑器：用于编辑规则列表，该规则列表定义了系统的行为。

模糊规则观察器和输出曲面观察器：用于观察(但不能编辑)模糊推理系统。它们是严格



图 2-2　模糊逻辑工具箱图形工具相互关系

的只读工具。模糊规则观察器是基于 MATLAB 模糊推理的图形显示。当模糊规则观察器被用于调试时，它能显示哪个规则正在运作，或各隶属函数的形状怎样影响结果，等等。输出曲面观察器用于显示输出对任何一个或两个输入的依赖性，即它为系统产生并且绘制一个输出曲面。

模糊逻辑工具箱的 5 个基本 GUI 工具之间能相互作用并交换信息。它们中的任何一个都能对工作区和磁盘进行读写（只读观察器同样能与工作区和（或）磁盘交换图形）。对于任何的模糊推理系统，这 5 个 GUI 工具中的任何一个或全部都是开放的。对于一个单独的系统，只要有一个以上的编辑器是打开的，则各 GUI 窗口都会知道其它窗口的存在，并且如果必要，它还能更新相关的 GUI 窗口。因此，如果隶属函数被隶属函数编辑器更名，那么这个变化将在模糊规则编辑器的规则中反映出来。对任意多个 FIS 系统，编辑器都可以同时打开。FIS 编辑器、隶属函数编辑器、模糊规则编辑器都能读并修改 FIS 的数据。但是，模糊规则观察器和输出曲面观察器在任何方式下都不能修改 FIS 的数据。

2.2.1　FIS 编辑器

如果已经存在一个模糊推理系统 aaa.fis，则在工作区中键入“fuzzy aaa”可以直接用模糊

逻辑工具箱打开这个模糊推理系统。

如果只是要打开 FIS 编辑器，则在工作区键入如下命令“fuzzy”。

打开的 FIS 编辑器如图 2－3 所示。

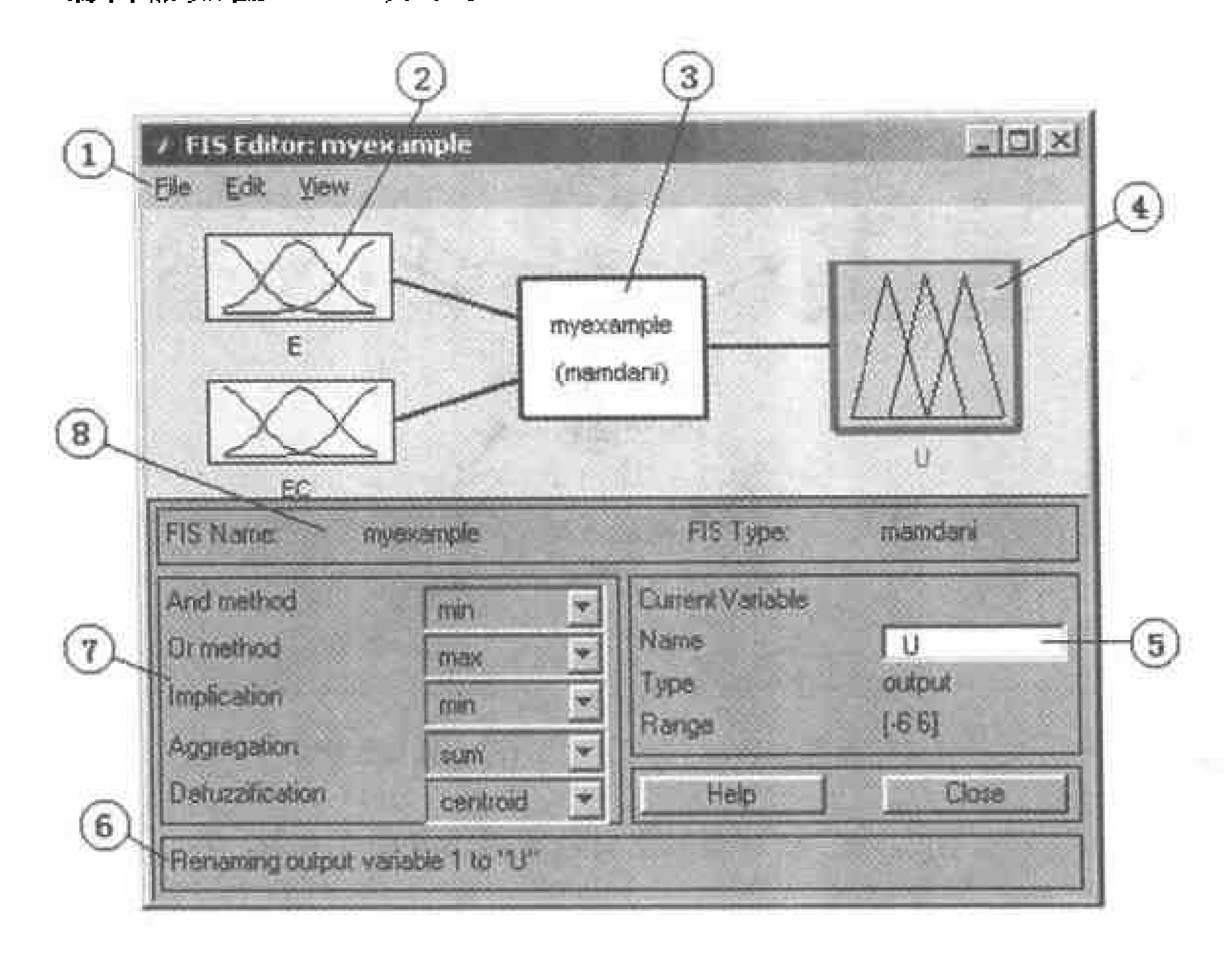


图 2－3　模糊推理系统编辑器

FIS 编辑器(如图 2－3 所示)中各部分的功能如下：

①——File(文件)菜单。该区域是每一个基本图形工具中都具有的 File 菜单项，在此可以进行打开、保存或编辑模糊系统的操作。

②——在该区域中，双击输入变量的图标，可打开隶属函数编辑器，以定义输入变量或输出变量的隶属函数。

③——在该区域中，双击系统示图，可打开模糊规则编辑器。

④——在该区域中，双击输出变量图标，可打开隶属函数编辑器，以定义输出变量或输入变量的隶属函数。

⑤——文本框。在该文本框中可对输入变量或输出变量进行命名或改名。

⑥——状态栏。在该状态栏显示上一步进行的操作。

⑦——下拉菜单。各下拉菜单用于选择模糊推理方法，例如选择“反模糊化”方法等等。

⑧——在该区域中显示系统名称。要改变系统名称，可在 File 菜单下选择“Save as ...”进行。

在系统的建立和修改过程中，FIS 编辑器中显示的图形将被即时刷新，包括系统名称、变量名、隶属函数等，以反映最新作出的修改。

2.2.2　隶属函数编辑器

下面定义与每个变量相关联的隶属函数。首先打开隶属函数编辑器。下面 3 种方式中的任何一个都可打开隶属函数编辑器。

①拉下 View 菜单项，选定"Edit Membership Functions..."。

②双击输出变量图标。

③在命令行键入"mfedit"。

打开的隶属函数编辑器如图 2－4 所示。

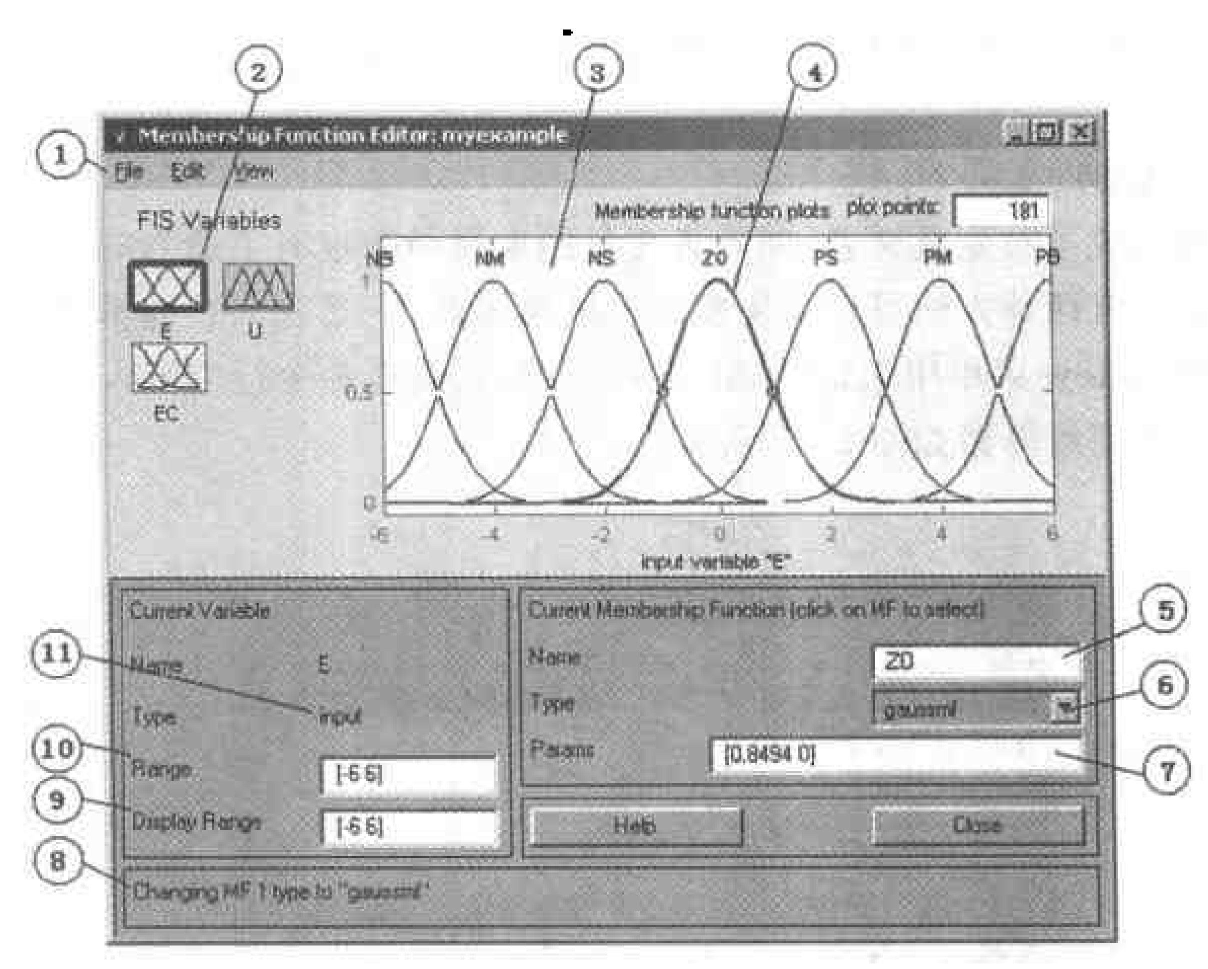


图 2－4　隶属函数编辑器

隶属函数编辑器中各区域的功能如下：

①——File(文件)菜单。在 5 个基本图形工具中的每一个 File 菜单下，均可进行打开、保存或编辑模糊系统的操作。

②——该区域为变量区，在此显示了所有已定义的输入变量和输出变量。单击某一变量，使其成为当前变量，就可编辑该变量的隶属函数。

③——该区域为绘图区。在此显示当前变量的所有隶属函数；

④——在该区域中，单击并选中一条隶属函数，就可以编辑该隶属函数的名称、类型、属性及参数。用鼠标按住一条隶属函数上的小圆圈并拖动鼠标，就可以改变所选中的隶属函数的形状及其参数，如果用鼠标按在隶属函数的小圆圈以外的曲线上并拖动鼠标，就可以平移该隶属函数的位置，而不改变其形状。

⑤——文本框。在该文本框中，可以改变当前选中的隶属函数的名称。

⑥——下拉菜单。在该下拉菜单中可以改变当前隶属函数的类型。

⑦——文本框。在该文本框中，可以改变当前选中的隶属函数的数字参数。

⑧——状态栏。在该状态栏显示上一步进行的操作。

⑨——文本框。在该文本框中，可设置当前图形的显示范围。

⑩——文本框。在该文本框中，可设置当前变量的范围。

⑪——在此区域，显示当前选中的变量的名称和类型。

2.2.3 模糊规则编辑器

在前面已经完成了对变量命名，同时各变量的隶属函数也有了适当的形状和名字，下面开始编辑模糊规则。有两种方法可以调用模糊规则编辑器，一是在 FIS 编辑器(同样也可在隶属函数编辑器)中的 View 菜单中选定“Edit rules...”，二是在命令行中键入“ruleedit”。

打开的模糊规则编辑器如图 2-5 所示。

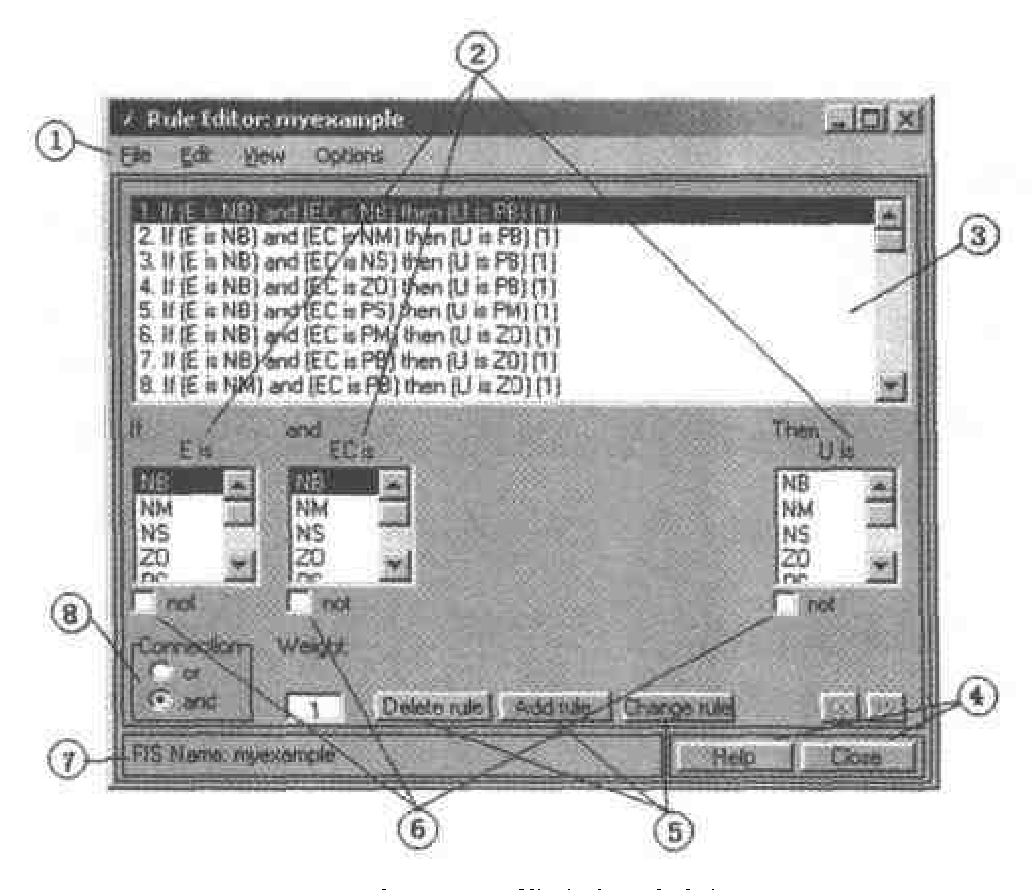


图 2-5　模糊规则编辑器

模糊规则编辑器各部分的功能如下：

①——File(文件)菜单。在 5 个基本图形工具中的每一个 File 菜单下，均可进行打开、保存或编辑模糊系统的操作。

②——输入输出量选择框。如前所述，输入输出的量是一个“词”而非数值。

③——模糊规则编辑器根据用户的操作而自动地书写或修改模糊规则，并将最新的模糊规则显示在该可视化区域。

④——Help 按钮和 Close 按钮。按 Help 按钮，可调用模糊规则编辑器的使用帮助；按 Close 按钮，则可关闭模糊规则编辑器。

⑤——Delete rule 按钮、Change rule 按钮和 Add rule 按钮。按 Delete rule 按钮，将选中的模糊规则删除；按 Change rule 按钮，可以修改选中的模糊规则；按 Add rule，将按照从输入输出菜单中所选的变量，建立新的模糊规则。

⑥——选中此复选框，将使模糊规则中输入输出的表述为“非”(取反)。

⑦——状态栏。在该状态栏显示上一步进行的操作。

⑧——选择该处的单选按钮，可以确定模糊规则中各输入间“与”或“或”的连接关系。

使用模糊规则编辑器构造规则非常容易。在 FIS 编辑器对输入输出变量进行定义和描述的基础上，可用模糊规则编辑器自动构造模糊规则语句。其方法是在模糊规则编辑器左边选择一个输入变量，并选中它的一个语言值(或称模糊子集、隶属函数)，然后在编辑器右边的输出变量框中选择一个输出变量，并选中它的一个语言值(或称模糊子集、隶属函数)，然后将这种联系添加到模糊规则中。对于一个规则中有多个输入变量，只要在输入变量之间再选择一个连接项(and 或 or)即可。如果在选择的变量下，选择 none 作为其隶属函数，则表示在这一规则中，这个变量不起作用。对任何变量，如果选择了下方的 not 复选框，都将对与该变量相关的性质取反。

从模糊规则编辑器的 Options 菜单下可以弹出 Format 菜单项，选择在 Format 菜单中的各项，可以看到模糊规则的不同格式。默认的是 verbose 项，此时看到的是模糊规则的详细格式。将其变成 symbolic，看到的模糊规则的显示没有太大不同，只是后者语言化色彩稍弱一些，因为它不依赖于“if ... then”这种格式。

不过如果将格式变换为 indexed，则模糊规则将以高度压缩的形式显示，其中不含任何自然语言。例如如下所示：

```
1  1  (1)  :  1
2  2  (1)  :  1
3  3  (1)  :  1
```

这种格式用于机器处理。结构中的首列对应于输入变量，第二列对应于输出变量，第三列表示应用每个规则的权值，第四列是一个简写，它指明规则是“OR”(2)还是“AND”(1)。在前两列中的数字指示隶属函数的索引数。

对规则 1 的文字解释是：如果输入 1 为 MF1（与输入 1 关联的第一个隶属函数），则输出 1 应当是具有权值为 1 的 MF1（与输出 1 关联的第一个隶属函数）。由于在此系统中仅有一个输入，故最后一列的“1”关联连接的“AND”是没有结果的。

在 Format 选项中，symbolic 格式与“if...then”项没有关系，索引格式甚至与变量名也没有关系。显然，系统功能不取决于变量和隶属函数叫什么名字。对变量命名的全部准则，就是为了用户能更方便地解释系统。因此一般而言，用 verbose 这一详细格式对变量命名会更方便。

应用以上三个模糊逻辑工具箱图形工具，就可以完整地建立模糊推理系统，要查看模糊推理系统的各种特性，需要应用下面两个观察器。

2.2.4 模糊规则观察器

模糊规则观察器的功能是可以令用户观察模糊推理图，并观察模糊推理系统的行为是否与预期的一样。

要打开模糊规则观察器，可在工作区中键入“ruleview”或从 FIS 编辑器（同样也可以从隶属函数编辑器或模糊规则编辑器）的 View 菜单中选择“View rules...”。

打开的模糊规则观察器如图 2－6 所示。

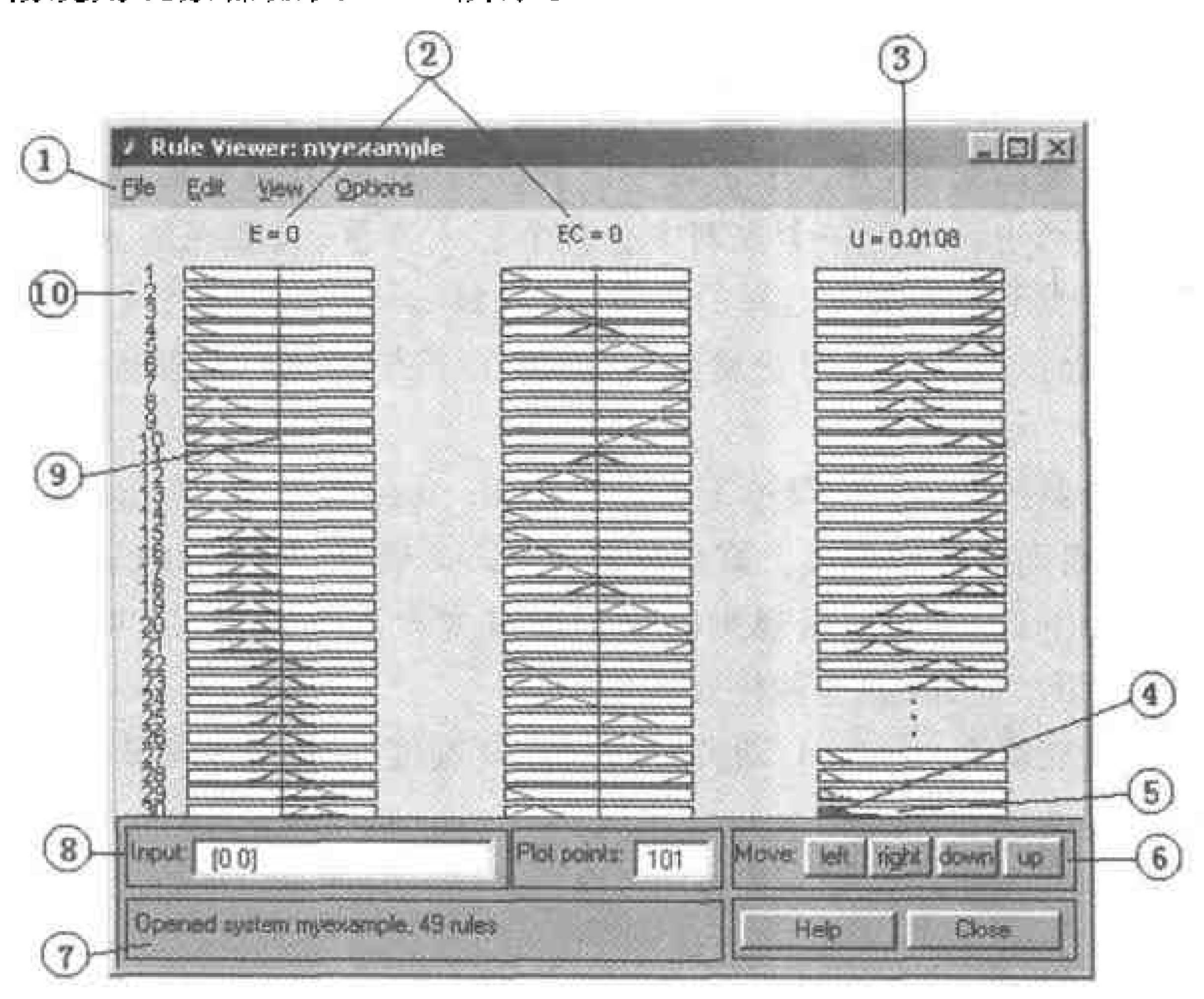


图 2－6 模糊规则观察器

模糊规则观察器各部分的功能如下：

① File(文件)菜单。在 5 个基本图形工具中的每一个"File"菜单下，均可进行打开、保存或编辑模糊系统的操作。

② 这两列黄色(默认色)的图形中，每列图形显示该列相应的输入变量如何应用在模糊规则中。在各列的顶端给出的是该输入变量的值。

③——在该列蓝色(默认色)的图形中，显示输出变量如何应用在模糊规则中。

④——由这条线给出了反模糊化的值。

⑤——该图显示各模糊规则的输出如何被结合，并构成聚类输出，然后再被反模糊化。

⑥——当系统很复杂、模糊规则个数很多时，整个计算机屏幕可能容纳不下所有模糊规则。在此的 4 个按钮可以将整个模糊观察器的图形区向上、下、左、右移动，以便于观察。

⑦ 状态栏。在该状态栏显示上一步进行的操作。

⑧——文本框。在该文本框中可设置具体的输入值。

⑨——用鼠标按住并移动此线，可改变输入值，并产生一个新的输出响应。

⑩——每一行图形表示一个模糊规则(图中有 30 个模糊规则)。单击模糊规则的标号 1，2 或 3，在状态栏中将显示该规则的表述。

模糊规则观察器显示了整个模糊推理过程的重要部分。它的基础是上一节中描述的模糊推理过程。上面的窗口图形中嵌套了 90 个小图。顶部横向的 3 个小图表示前提条件与第一个规则的结果。每一个规则与一行图形相对应，每一列上是一个变量。前两列的黄色小图显示的是前提(即每个规则的"if"部分)引用的隶属函数。第三列的蓝色小图显示的是结果(即每个规则的"then"部分)引用的隶属函数。单击图形最左边的某个规则标号 1，2 或 3，将在图形下端显示相应的规则。若在某输入变量下有一个空白小图，则表示相应模糊规则中对该输入变量选择的描述为 none。第三列的最后一个图形中表示对给定推理系统的总和加权判定。对系统的这一判定将取决于输入值。

从模糊规则观察器的显示中可以立即解释全部模糊推理过程。模糊规则观察器同样显示了某一隶属函数的形状会如何影响整体结果。由于模糊规则观察器中画出了每一规则每一部分的图形，因此当系统特别大时，它会显得比较"笨"。但随着计算机速度和内存的不断发展与扩充，只要输入输出的数目不是太多、模糊规则的数量不是特别大，模糊规则观察器都能轻易地完成其任务。

2.2.5 输出曲面观察器

模糊规则观察器非常详细地显示了在某一时刻的计算。在这个意义上，它给出的是模糊推理系统的微观视图。但如果要看到模糊推理系统的全部输出曲面，也就是与整个输入区间相对应的整个输出区间，则需要打开模糊逻辑工具箱的最后一个 GUI 工具，即输出曲面观察

器。

要打开输出曲面观察器，可在工作区中键入“surfview”或从 FIS 编辑器(同样也可以从隶属函数编辑器、模糊规则编辑器或模糊规则观察器)的 View 菜单中选择“View surface...”。

打开的输出曲面观察器如图 2－7 所示。

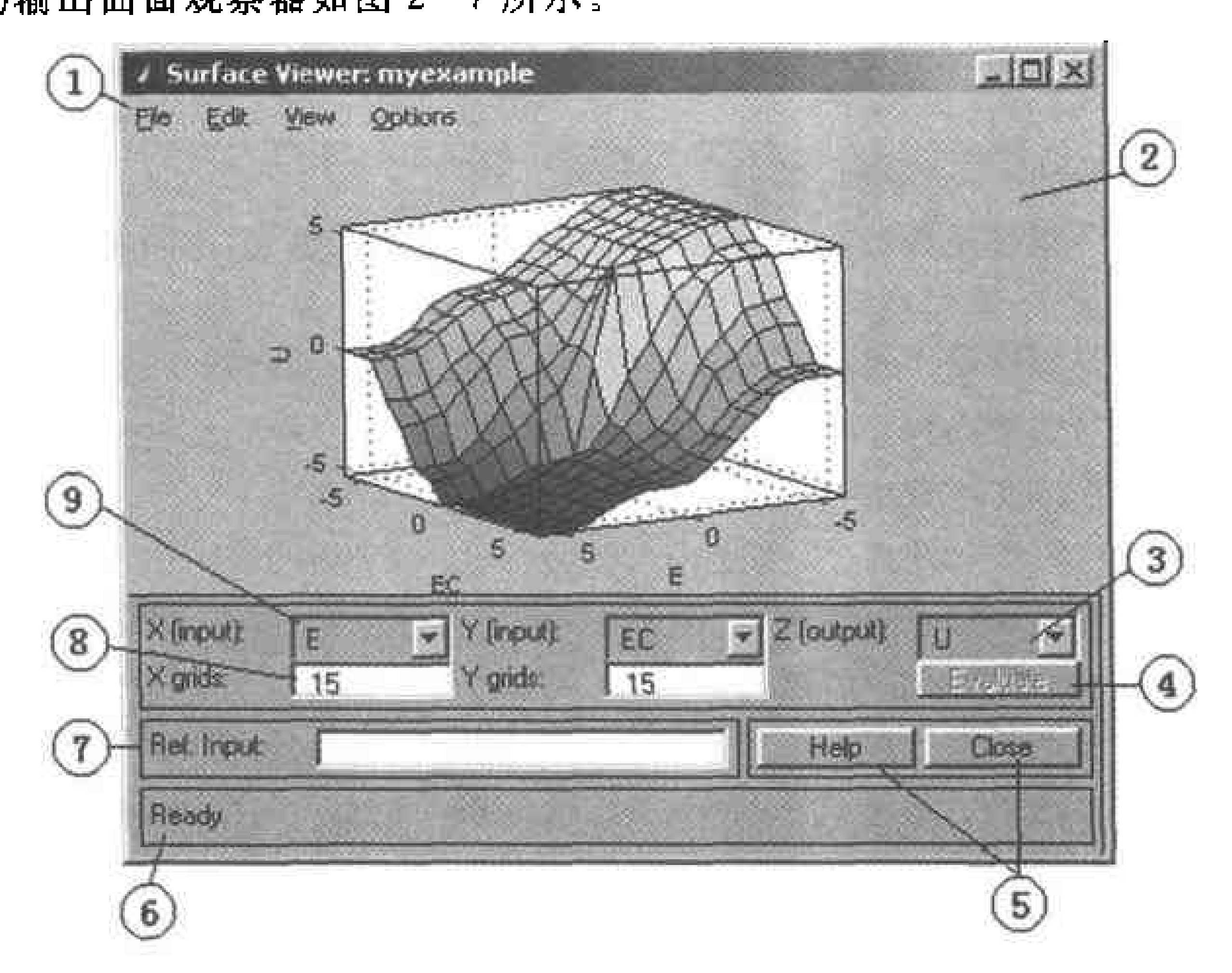


图 2－7　输出曲面观察器

模糊观察器各部分的功能如下：

①——File(文件)菜单。在 5 个基本图形工具中的每一个 File 菜单下，均可进行打开、保存或编辑模糊系统的操作。

②——输出曲面。在输出曲面中显示了系统的任一输出与任一或两个输入间的对应关系。在输出曲面上拖动鼠标，可以转动坐标轴，以便于观察输出曲面。

③——下拉菜单。在该下拉菜单中可指定要显示的输出变量。

④——Evaluete 按钮。在设置了新的输入变量之后，单击此按钮，可计算新的输出曲面并绘出其图形。

⑤——Help 按钮和 Close 接钮。按 Help 按钮，可调用输出曲面观察器的使用帮助；按 Close 按钮，则可关闭输出曲面观察器。

⑥——状态栏。在该状态栏显示上一步进行的操作。

⑦——在输出曲面上只能显示两个输入变量。当输入变量的个数多于两个时，在该文本

框中可设置未在输出曲面中显示出的输入变量的值。

⑧——文本框。在该文本框中可指定对输入空间绘图的网格数。

⑨——下拉菜单。在该下拉菜单中可指定要在输出曲面中显示的一或两个输入变量。

在输出曲面观察器中，对于单输入单输出情形，可以在一个图形中看到全部映射；对于二输入单输出情形，因为MATLAB可以轻易地完成三维图形的绘制，所以系统同样能完成工作。当系统的输入输出个数之和超过三个（三维）时，则难以作出输出的图形。因此，在输出曲面观察器的下端设置了弹出式选择框，从中可在多个变量中任选两个输入作为三维图形中X，Y轴的输入，并以输出作为Z轴的输入。在变量选择框正下方是两个文字输入区，在其中输入合适的数字，则可以在X轴和Y轴上设置相应数目的网格线。对于复杂问题，将网格线数目设置相对小一些，可以使计算时间更理想一些。在改变了变量和网格的设置后，按Evaluate按钮，系统开始计算，并在计算完成后立即绘制结果图形。在看到输出曲面后，要更改X轴或Y轴的网格，只要在文字区中更改数字，然后单击输出曲面上的X轴或Y轴，系统会立即根据新数字重绘图形。

输出曲面观察器变量和网格设置区的下面是Ref. Input区。假定现在有一个四输入单输出系统并想观察其输出曲面，则输出曲面观察器可以产生一个三维的输出曲面，其中两维是四个输入变量中的任意两个，这两个输入是可变的。但由于计算机无法显示五维图形，所以四个输入变量中的另外两个必须是定值。在这种情况下，输入为四维矢量。但其中用NaN(not a number)占据可变输入的位置，同时将其值保持固定的输入用一个数字在Ref. Input区中给出。

本节中介绍了模糊逻辑工具箱中5个主要的GUI工具。对于一个简单的系统，用户可能会认为使用模糊逻辑工具箱显得有些麻烦。但只要用户掌握了模糊逻辑工具箱的使用，则不论面对的是简单的系统还是一整类相似的决策问题，模糊逻辑工具箱都会发挥其强大的功能。有了它提供的方便，就可以迅速修改任何一个系统。

注意，如果要将建立或修改后的模糊推理系统保存到以后使用，则应当将其保存到磁盘，而不是仅保存到MATLAB工作区。当将模糊系统存到磁盘时，用户存入的是一个用ASCII码表示该系统的FIS文件，文件属名为.fis。这是一个文本文件，它简单易懂，并可被编辑或修改。如果要将模糊系统保存到MATLAB工作区，那么用户需要建立一个变量（变量名由用户指定），这个变量将FIS系统作为MATLAB的一个结构进行工作。.fis文件和FIS结构所表示的是同一个系统。

2.2.6　自定义模糊推理系统

如果用户想在使用模糊逻辑工具箱时将自定义函数包括进去，那么需要遵循下面的规定，即用户提供的AND方法、OR方法、聚类方法和反模糊化方法需要与MATLAB中的max，

min 或 prod 的工作方式相似，也就是说他们必须能向下运算矩阵的列。例如，与 min 函数相似，蕴涵应当对矩阵一列一列地运算，如下例：

```
a=[1 2; 3 4];
b=[2 2; 2 2];
min(a,b)
ans =
1 2
2 2
```

用户可以用 M 文件建立自己的隶属函数。这些函数的取值必须在 0 到 1 之间。对自定义隶属函数的限制是其参数不能超过 16 个。

例如，假定要建立一个名为“custmf”的自定义隶属函数，则步骤如下：

①为函数建立一个 M 文件 custmf.m，其取值在 0 到 1 之间，且参数最多为 16 个。

②在隶属函数编辑器的 Edit 菜单选择“Add Custom MF...”，弹出一个对话框。

③在弹出对话框的 M-file function name 文本框中输入自定义隶属函数的 M 文件名“custmf”。

④在 Parameter list 文本框中输入在自定义隶属函数中想要使用的参数矢量。

⑤在 MF name 文本框中对自定义隶属函数指定一个名称。这个名称必须与模糊推理系统中使用的其它任何隶属函数的名称都不相同。

⑥单击该弹出对话框的“OK”按钮。

用户可以将这个文件命名为“testmf1.m”，并用用户选择的参数将其载入隶属函数编辑器中。

2.3 模糊逻辑工具箱的命令行工作方式

2.3.1 系统结构函数

系统结构函数主要有：readfis，setfis，getfis，showfis，structure。

要载入一个已存在的名为“aaa.fis”的模糊推理系统，须在工作区中键入

```
a=readfis('aaa.fis')
```

MATLAB 工作区将输出该模糊推理系统的如下信息：name(系统名称)，type(推理类型)，andMethod(与算子)，orMethod(或算子)，defuzzMethod(反模糊化方法)，impMethod(蕴涵算子)，aggMethod(聚类算子)，input(输入向量维数)，output(输出向量维数)，rule(模糊规则个数)。

函数 getfis(a)返回的信息几乎与结构 a 的信息完全一样。关于函数 getfis(a)的使用请参

见后面的函数详解。

与函数 getfis 对应的函数是 setfis。使用该函数可以改变 FIS 的任何属性。例如，要改变系统的名称，可以键入

a = setfis(a,'name','gratuity');

则将系统名称设置为 gratuity。

使用函数 showfis(a)可以更深入地看到 FIS 的结构。showfis(a)的主要功能是用于调试，但它能给出 FIS 结构中记录的所有信息。

当变量 a 被指定为某一个模糊推理系统时，在命令行中可以直接调用该模糊推理系统 5 个 GUI 的任何部分。下列函数中的每一个都可以调出关联到 GUI 的模糊系统。

fuzzy(a)：调出 FIS 编辑器

mfedit(a)：调出隶属函数编辑器

ruleedit(a)：调出模糊规则编辑器

ruleview(a)：调出模糊规则观察器

surfview(a)：调出输出曲面观察器

另外，如果 a 是一个 Sugeno 型 FIS，则 anfisedit(a)将调出 ANFIS 的图形化工具。

只要打开了任何一个 GUI，就可以通过它的下拉菜单打开其它的 GUI，而不必再用命令行方式。

2.3.2　系统显示函数

模糊逻辑工具箱在命令行中设计了 3 个函数，可以用它们得到高水准的模糊推理系统图形。这 3 个函数是 plotfis，plotmf 和 gensurf。

第一个函数绘制出与 FIS 编辑器上相同的全系统模块图，其使用格式为

plotfis(a)

如果所有的 MATLAB 图形或 GUI 窗口都已被关闭，函数 plotmf 将画出与给定变量相关联的所有隶属函数。其使用格式为

plotmf(a,'input',1)

plotmf(a,'output',1)

如果 5 个图形化工具中的任何一个是打开的，那么调用 plotmf 后，隶属函数图将以 GUI 或 MATLAB 图的方式显示。

第三个函数 gensurf 的功能，是画出给定系统输入变量中的任一个或两个与输出变量中的任一个之间的图形，该图形是二维曲线或三维曲面。注意当有三个或更多个输入时，gensurf 只能产生两个输入为可变而其余输入为固定的图形。这种情况与前面介绍的 genfis 是一样的。gensurf 的使用格式为

gensurf(a)

2.3.3 在命令行中建立系统

事实上，完全可以在没有 GUI 工具的情况下使用模糊逻辑工具箱。例如，要完全从命令行中建立小费计算系统，可以使用函数 newfis，addvar，addmf 和 addrule。这些命令的使用方法参见后面的函数详解。在此给出一个简单的例子。在 MATLAB 命令行中键入如下的命令：

```
a=newfis('tipper');
a.input(1).name='service';
a.input(1).range=[0 10];
a.input(1).mf(1).name='poor';
a.input(1).mf(1).type='gaussmf';
a.input(1).mf(1).params=[1.5 0];
a.input(1).mf(2).name='good';
a.input(1).mf(2).type='gaussmf';
a.input(1).mf(2).params=[1.5 5];
a.input(1).mf(3).name='excellent';
a.input(1).mf(3).type='gaussmf';
a.input(1).mf(3).params=[1.5 10];
a.input(2).name='food';
a.input(2).range=[0 10];
a.input(2).mf(1).name='rancid';
a.input(2).mf(1).type='trapmf';
a.input(2).mf(1).params=[-2 0 1 3];
a.input(2).mf(2).name='delicious';
a.input(2).mf(2).type='trapmf';
a.input(2).mf(2).params=[7 9 10 12];
a.output(1).name='tip';
a.output(1).range=[0 30];
a.output(1).mf(1).name='cheap'
a.output(1).mf(1).type='trimf';
a.output(1).mf(1).params=[0 5 10];
a.output(1).mf(2).name='average';
```

```
a.output(1).mf(2).type='trimf';
a.output(1).mf(2).params=[10 15 20];
a.output(1).mf(3).name='generous';
a.output(1).mf(3).type='trimf';
a.output(1).mf(3).params=[20 25 30];
a.rule(1).antecedent=[1 1];
a.rule(1).consequent=[1];
a.rule(1).weight=1;
a.rule(1).connection=2;
a.rule(2).antecedent=[2 0];
a.rule(2).consequent=[2];
a.rule(2).weight=1;
a.rule(2).connection=1;
a.rule(3).antecedent=[3 2];
a.rule(3).consequent=[3];
a.rule(3).weight=1;
a.rule(3).connection=2
```

则建立了一个名为 tipper 的模糊推理系统。

在这个过程中，最困难的部分是建立模糊推理系统使用的模糊规则的快速方法。这项工作由函数 addrule 完成。

每个输入或输出变量都有一个指针，每个隶属函数也有一个指针。在语句表述中建立规则的方式如下：

if input1 is MF1 or input2 is MF3 then output1 is MF2 (weight = 0.5)

规则根据下列逻辑返回到结构中：如果系统有 m 个输入和 n 个输出，则规则结构中的前 m 个向量元素与第1到 m 个输入相对应，第一列中的元素是与输入1相关的隶属函数指针，第二列中的元素是与输入2相关的隶属函数指针，以此类推。其后的 n 列以相同方式关联到输出。第 $m+n-1$ 列是与规则相关的权（通常为1），第 $m+n+2$ 列指定其连接方式（AND=1，OR=2）。因此，与上一个模糊规则相关联的结构为

1 3 2 0.5 2

2.3.4 FIS 求解

要在给定输入下求模糊系统的输出，可以使用函数 evalfis。

例如，下列语句求出了输入为[1 2]时的小费额。

```
a = readfis('tipper');
evalfis([1  2],a)
```

结果为

ans = 5.5586

因为不同的输入向量表示在输入结构的不同部分当中，所以这个函数也可用于在多重输入集情况下计算模糊推理系统的输出。一次性地完成多重计算，可以极大地提高计算速度。例如 evalfis([3 5; 2 7],a)

结果为

```
ans =
12.2184
7.7885
```

2.3.5 FIS 结构

模糊推理结构是一个 MATLAB 对象，它包含了模糊推理系统的所有信息。这个结构存储在每个 GUI 工具中。使用 getfis，setfis 等函数可以方便地检查该结构。同样，也可以用 structure.field 语法查看该结构（参阅"模糊逻辑工具箱的命令行工作方式"）。

一个给定的模糊推理系统的所有信息都包含在 FIS 结构中，包括变量名、隶属函数定义等。这个结构本身可以视为一个层级结构，如图 2-8 所示。

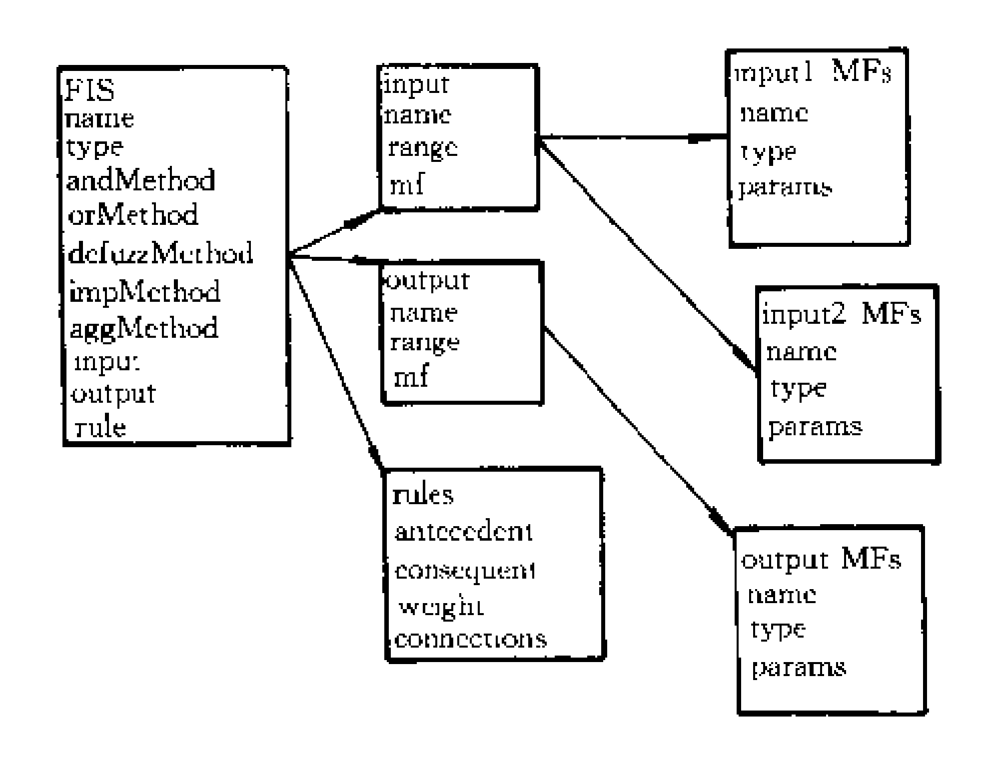


图 2-8 模糊推理系统结构

在 MATLAB 命令行中键入"showfis(a)"，则将给出 FIS 的所有信息列表。

与FIS结构相关的命令行函数包括：getfis、setfis，showfis、addvar、addmf，addrule，rmvar和rmmf。

将FIS文件存储到磁盘时，使用了一个特定的文本文件格式.fis。可用函数readfis和writefis对这些文件进行读写。

如果需要，用户还可以通过任何文本编辑器来编辑文件.fis以修改FIS，而不必使用任何的GUI工具。但要注意的是，在文本编辑中，只要改变了一项，就必须改变其它项。例如，如果以编辑文本文件的方法删除了一个隶属函数，用户就必须同时删除所有的需要该隶属函数的模糊规则。

2.4 模糊推理系统类型的转换

MATLAB命令行函数mam2sug可以将Mamdani系统转换成带常量输出隶属函数的Sugeno系统（不必是单输出）。它使用与Mamdani系统所有输出隶属函数相关联的重心。参见附录中的函数详解。

2.5 神经-模糊推理编辑器ANFIS

到目前为止，我们所见到的模糊推理系统的基本结构是将输入特性映射到输入隶属函数上，再将输入隶属函数映射到模糊规则上，然后将模糊规则映射到输出特征集上，再将输出特征映射到输出隶属函数上，最后将输出隶属函数映射到一个单值输出上或一个与输出相关的决策上。在其中，我们仅考虑了确定的隶属函数，并在选择隶属函数时带有一定的主观性。而模糊推理的应用对象，是用户对模型变量的特性已经预先决定了的系统。

但有时候，因为要在某个系统中应用模糊推理，为此采集了许多输入输出数据，并想将这些数据用于建模、模型跟踪或其它方案。这时候，往往并没有一个基于系统变量特征的先验模型结构，而且在某些条件下，模糊控制器的设计者无法一边看着数据，一边就能够弄清楚隶属函数的形状。在这种情况下，可以用为输入输出数据选择适当的参数（而不是任意地选择与给定隶属函数相关的参数）来调整隶属函数，以说明数据的变化类型。在模糊逻辑工具箱中的anfis函数中嵌入的神经自适应训练技术即可完成该工作。

ANFIS是adaptive neuro－fuzzy inference system（自适应神经－模糊推理系统）的缩写。

工具箱函数anfis使用一个给定的输入输出数据集构造出一个模糊推理系统，并用一个单独的反向传播算法或该算法与最小二乘相结合的方法来完成对系统隶属函数参数的调节。这使得模糊系统可以从其建模数据中学习信息。在MATLAB中有命令行和图形用户界面两种方式可以应用工具箱函数anfis。其图形用户界面称为ANFIS编辑器。由于anfis的命令行

方式比较麻烦，在此仅对 ANFIS 图形界面编辑器作一介绍。

ANFIS 编辑器将模糊推理技术用于数据建模。与在其它的模糊推理工具中一样，隶属函数的形状由其参数决定，改变这些参数就将改变隶属函数的形状。

2.5.1 神经-模糊推理

anfis 既可从命令行中访问，也可通过 ANFIS 图形编辑器访问。由于 anfis 在命令行中的功能与在 ANFIS 图形界面编辑器中的功能相同，因此在本章的讨论中，它们在某种程度上是互换使用的，除非我们通过对图形界面的说明将它们区分开来。

输入输出之间的映射关系可用与神经网络结构相似的网络型结构来解释，即它是先通过输入隶属函数及其参数、然后再通过输出隶属函数及其参数，完成从输入到输出的映射。通过训练过程，可以改变与隶属函数相关的参数。在估计隶属函数参数时，anfis 使用的是反向传播算法或反向传播算法与最小二乘估计相结合的方法。

anfis 使用的建模方法与系统辨识的方法相似。首先假定一个参数化的模型结构（将输入关联到隶属函数、规则、输出隶属函数，等等），然后采集输入输出数据。再使用 anfis 训练 FIS 模型，根据某个选定的误差准则修正隶属函数参数，使得 FIS 系统模仿提供给系统的训练数据。

一般地，如果输入 anfis 的训练数据能够完全地表示将要建模的系统的特征，那么得到的 FIS 模型工作得非常好。但在实际中数据采集时往往含有噪声，而训练数据无法表示要建模的系统的全部特征。此时需要用检验数据和测试数据进行模型确认的工作。

在用 ANFIS 编辑器进行模型确认过程中，将输入输出数据中未经训练过的输入数据传送到被训练的 FIS 模型中，观察 FIS 模型的输出数据。在 anfis 中，也可以使用另一类型的数据集即检验数据集进行模型确认，其作用是控制 FIS 模型过度匹配数据的可能性。当检验数据与训练数据被同时输入 anfis 时，则可选定 FIS 模型，其参数为使检验数据模型误差最小时的参数。

模型确认中重要的是要选择一个良好的输入输出数据集，这个数据集既要能典型地代表被训练模型要模仿的数据集，又要与训练数据集有明显的区别，以免确定过程失去价值。

在模型确认中使用检验数据集的基本思想，是训练中的某一时刻之后，模型开始完全匹配训练数据集。原则上，当训练到完全匹配这一点发生时，对检验数据的模型误差将趋于减少，然后由于模型过度地匹配数据，检验数据的模型误差会突然增加。

一般而言，anfis 远比前面讨论的模糊推理系统复杂，而且它并不是对所有的模糊推理系统都适用。具体地说，anfis 只支持 Sugeno 型系统，而且必须是各规则的权均为 1 的一阶或零阶 Sugeno 型系统，并且系统是单输出，其输出值由加权平均反模糊化方法（即线性或常量输出隶属函数）获得。

2.5.2　ANFIS 编辑器

要使用 ANFIS　图形界面编辑器，可在命令行中键入“anfisedit”，屏幕上将出现如图 2－9 所示的图形界面。

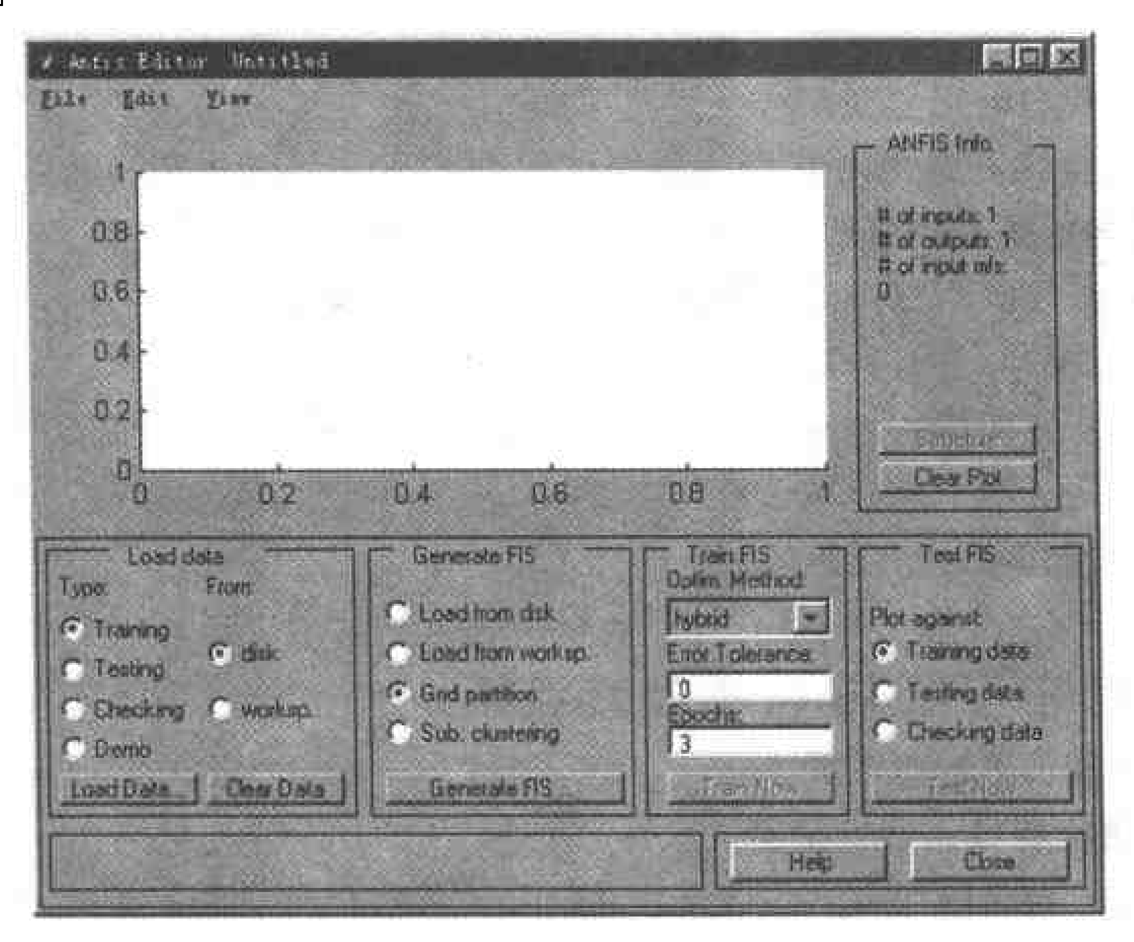


图 2－9　ANFIS 编辑器

在这个图形工具中可以完成如下工作：

① 载入数据(训练、测试、检验)。在 GUI 的“Load data”部分选择合适的单选按钮，然后选择“Load Data...”，在图形区上会画出载入的数据。

② 产生或载入一个初始化 FIS 模型。在 GUI 中使用“Generate FIS”选项。

③ 观察 FIS 模型结构。当初始化 FIS 模型产生或载入后，选择“Structure”按钮。

④ 选择 FIS 模型参数优化方法。可选择反向传播算法或反向传播算法与最小二乘法的结合(混合法)。

⑤ 选择训练时长和训练误差容忍限。

⑥ 训练 FIS 模型。选择“Train Now”按钮。在训练中将调整隶属函数参数并在图形区画出训练数据(和/或检验数据)误差图。

⑦ 观察相对于训练、检验或测试数据输出的 FIS 模型输出。选择“Test Now”按钮。该函数在图形区绘制对 FIS 输出的测试数据。

用 ANFIS 编辑器可以载入训练初始化 FIS、保存训练过的 FIS、新建一个 Sugeno 系统以及打开其它图形工具来描述训练过的 FIS 模型。

要用 anfis 或 ANFIS 图形界面编辑器训练一个模糊推理系统，首先需要一个训练数据集，在训练数据集中应当包含要建模的目标系统所期望的输入输出数据对。有时候用户可能还需要一个可选测试数据集，它能检验结果模糊推理系统的归纳能力，或(和)需要一个检验数据集，它对训练期间模型的完全匹配将会有所帮助。

如前所述，完全匹配是由对在训练数据上训练的 FIS 系统进行检验数据测试来估计的。如果它们的误差表示模型完全匹配，则选择那些与最小检验误差相关联的隶属函数参数。在作出此决定之前，必须相当仔细地检验训练误差图形。通常是在观察目标系统的基础上采集这些训练和检查数据集，并且将数据集分别存储于不同的文件中。

需要注意的是，任何载入 ANFIS 图形界面编辑器的数据集(或在命令行函数 anfis 中应用的数据集)都必须是一个矩阵。在该矩阵中，除了最后一列外，都必须是一个以矢量形式排列的输入数据。输出数据必须在最后一列。

2.5.3 应用 ANFIS 编辑器的步骤

应用 ANFIS 编辑器的步骤如下：

1. 第 1 步：用检验数据确定模型

键入“anfisedit”打开 ANFIS 编辑器。要载入训练数据集，须选择 ANFIS 编辑器左下的“Load Data”区中“Type”下的单选按钮“Training”和“From”下的单选按钮“disk”，然后单击“Load Data”框底部的“Load Data...”。从数据集所在的目录下载入训练数据文件(例如 fuzex1trnData.dat)。在 ANFIS 编辑器中心的图形区会将训练数据以“O”的形式绘制出来，如图 2-10 所示。

注意横轴标号是 data set index，它给出了横向标记，从它可以得到输入数据的值(不论输入是向量还是标量)。

然后，在编辑器“Load data”区的“Type”列选择“Checking”，或者在“From”下选择“disk”，从检验数据集所在的目录下载入数据，则检验数据将以“+”的形式叠加在编辑器图形区的训练数据上，如图 2-10 所示。

这个数据集用于调整数据建模的最佳隶属函数参数从而训练模型系统。下一步是为 anfis 的训练指定一个初始模糊推理系统。

用户可以按自己的经验初始化模糊推理系统的参数，但如果对系统参数没有任何先验的认识，则可由 anfis 来完成初始化。

用 anfis 初始化模糊推理系统的步骤是：

① 在 ANFIS 编辑器的“Generate FIS”区中选择缺省分割法“Grid partition”。

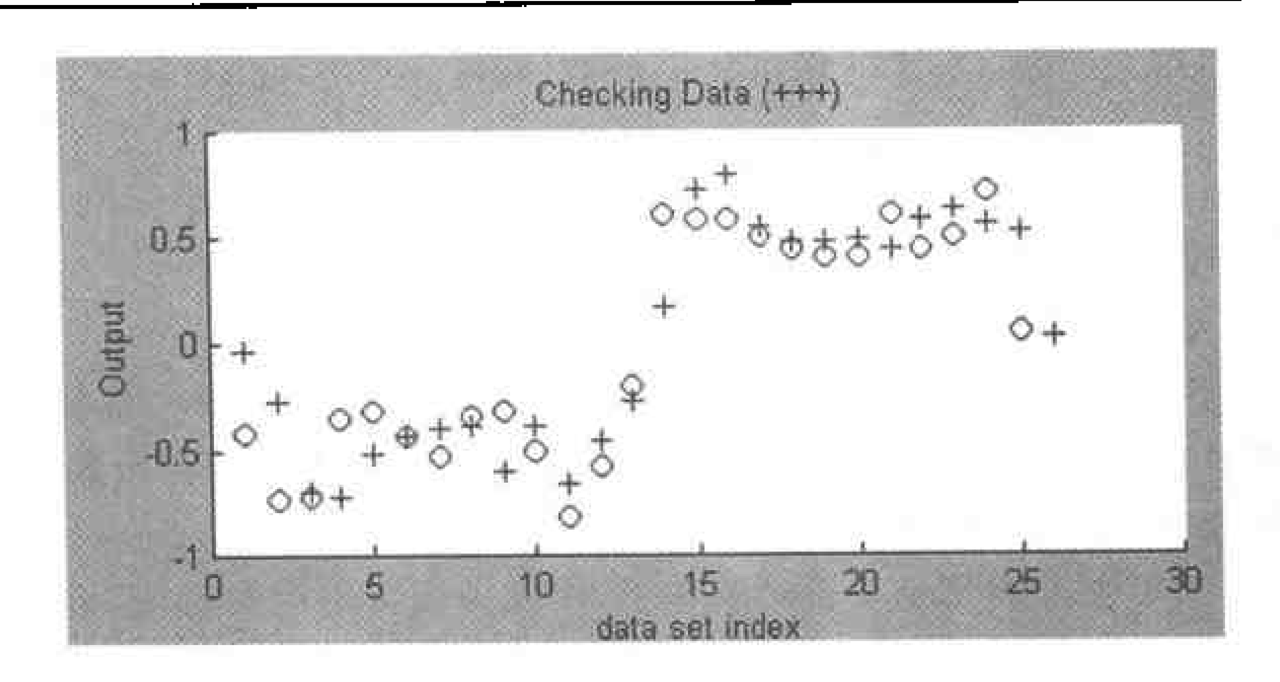


图 2－10　训练数据与检验数据图形

② 单击“Generate FIS”区底部的“Generate FIS...”按钮，它弹出一个菜单，从中用户可以选择输入和输出隶属函数的类型及输入隶属函数的个数。注意在输出隶属函数中只有两个选项：constant 和 linear，其原因是 anfis 只工作于 Sugeno 型系统中。

③ 在图 2－11 中填入输入变量个数，然后单击“OK”按钮。在 ANFIS 编辑器的底端将出现“a new fis generated”的信息。

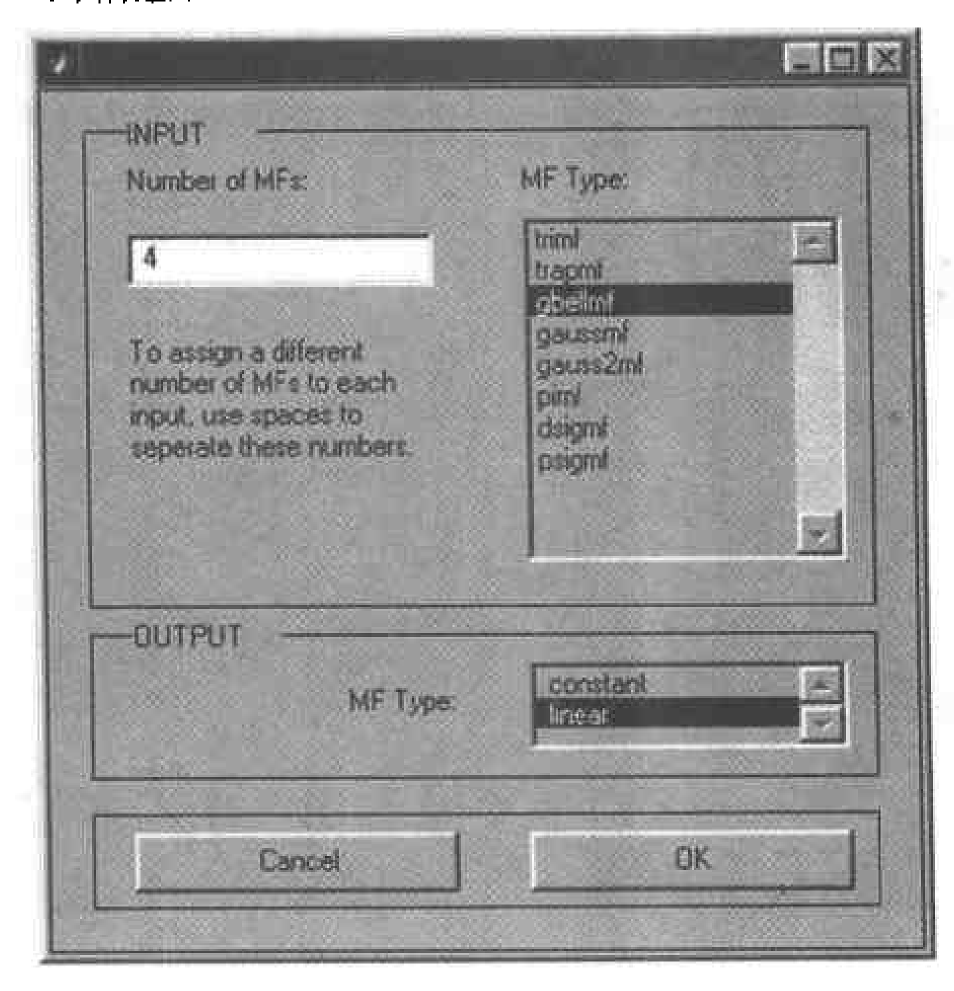


图 2－11　定义神经-模糊系统的输入输出隶属函数

同样，用户也可以在命令行中使用函数 genfis1（网格分割法）或 genfis2（消去聚类法）产生 FIS。

尽管 MATLAB 并不希望用户选择自己的隶属函数，但用户仍可在 anfis 中选择想要的具有特定参数的隶属函数，以初始化 FIS 并开始训练。

自定义 FIS 结构与参数的步骤是：

① 从 ANFIS 编辑器的 View 菜单打开“Edit membership functions”菜单项。

② 添加期望的隶属函数（对 anfis，自定义函数选项无效）。输出隶属函数必须都是常量或都为线性的（请参阅有关“FIS 编辑器”和“隶属函数编辑器”的章节）。

③ 在 View 菜单中选择“Edit rules”菜单项，用规则编辑器产生规则（请参阅“规则编辑器”）。

④ 在 View 菜单中选择“Edit FIS properties”菜单项，对用户的模糊推理系统命名，并将其保存到工作区或磁盘中。

⑤ 使用 View 菜单回到 ANFIS 图形界面编辑器，开始训练 FIS。

⑥ 要为 ANFIS 的初始化载入一个已存在的 FIS，请在 ANFIS 编辑器下中部的“Generate FIS”部分中单击“Load from worksp”或“Load from disk”（如果要使用的 FIS 已经预先存入磁盘，则从磁盘中载入 FIS，否则从工作区载入用户的 FIS），单击其中之一，载入用户的 FIS。

生成 FIS 后，单击 GUI 右上“ANFIS info”区中的“Structure”按钮，可以观察模型结构。此时将弹出一个新的图形工具，如图 2-12 所示。

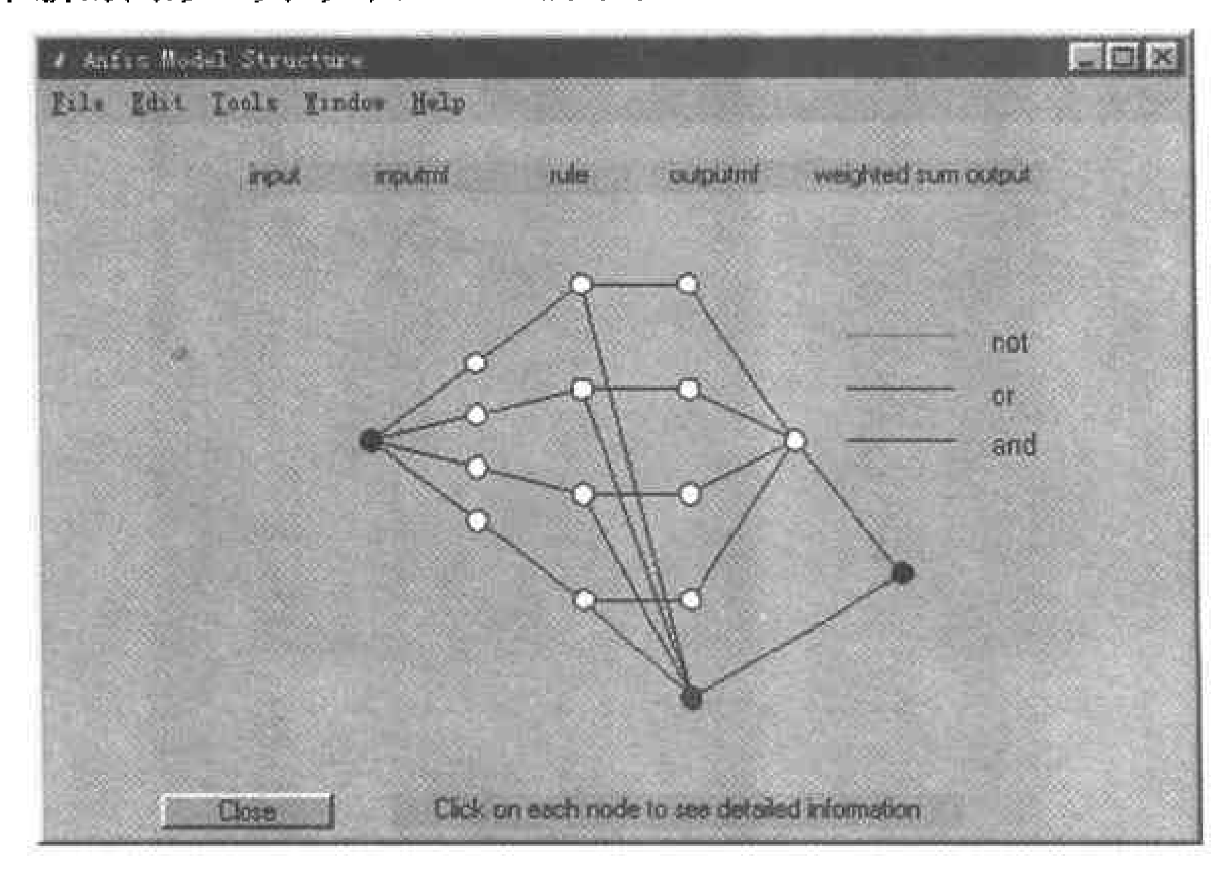


图 2-12　模糊推理系统的结构

在节点图的分支上是颜色编码，它指明在规则中使用的是 AND，OR 还是 NOT。在节点上单击，将显示结构信息。

2. 第 2 步：anfis 训练

在模糊推理系统的训练中，有两种 anfis 参数优化方法选项，即 hybrid（默认值，它是反向传播算法与最小二乘法的结合，即混合法）和 backpropa（反向传播算法）。Error Tolerance（误差容许限）用于建立一个停止训练的标准，它与误差的大小有关。当训练数据误差保持在容忍限之内时，则停止训练。如果不知道训练误差的范围，最好将其设为 0。

要开始训练，可按照如下步骤进行：

① 在 ANFIS 编辑器右下的"Train FIS"区下的"Optim Method"中，选择优化方法为"hybrid"。

② 在"Train FIS"区下的"Error Tolerance"设定误差容忍限。

③ 在"Train FIS"区下的"Epochs"将训练时间设为"40"（默认值是 3）。

④ 选定"Train Now"按钮。

屏幕上的显示如图 2-13 所示。

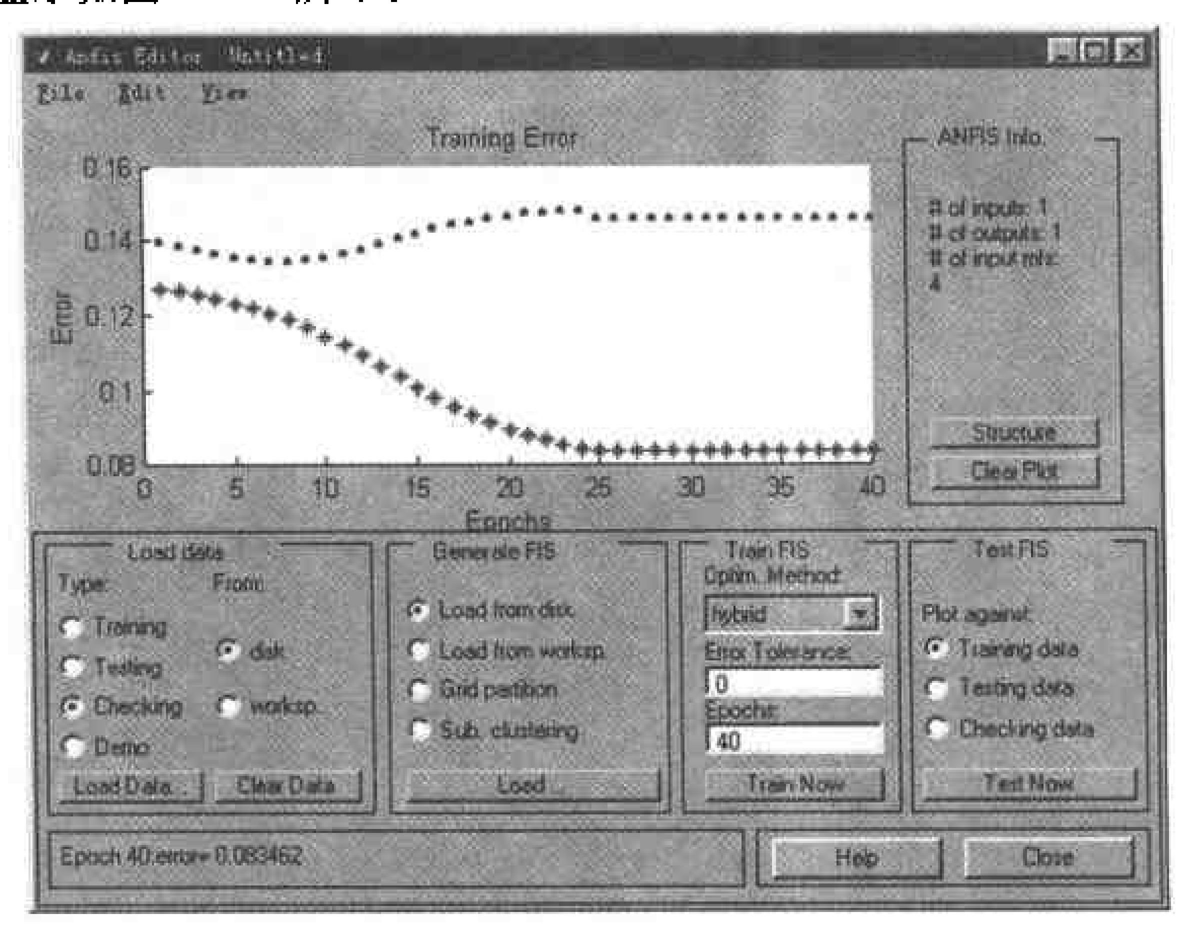


图 2-13　anfis 训练结果

注意在训练中检验误差是怎样减少至某一个点，然后再跳跃上去的。这个跳跃点就表示模型完全匹配的点。anfis 将选择与最小检验误差（就在跳跃点之前）相关联的隶属函数参数。这个例子说明 anfis 的检验数据选项是有用的。

3. 第 3 步：用训练过的 FIS 测试用户的数据

要用检验数据测试模糊推理系统，请在 ANFIS 编辑器右下方的"Test FIS"部分选中"Checking data"，然后单击"Test Now"按钮。现在对 FIS 测试检查数据，系统工作良好。检验结果如图 2-14所示。

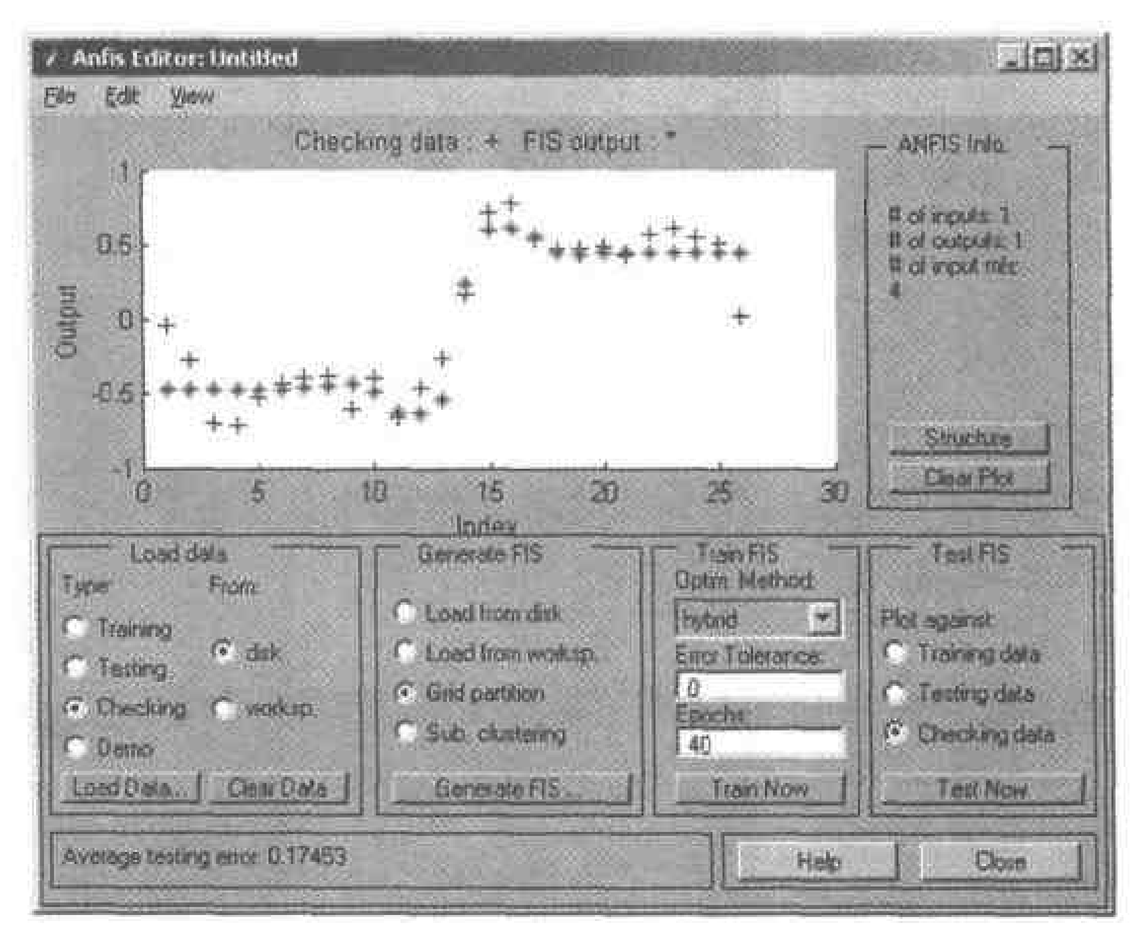


图 2-14　anfis 检验结果

当训练数据集和检验数据集充分不同时，系统结果将不会像上面进行的那样良好，即训练过的 FIS 没有很好地提取出数据集的特征。在这种情况下，用户或者需要在训练中选择更多数据，或者需要修改选择的隶属函数(包括隶属函数的数目和类型)。



第3章 模糊 PID 控制

PID 控制具有结构简单、稳定性能好、可靠性高等优点，尤其适用于可建立精确数学模型的确定性控制系统。在控制理论和技术飞速发展的今天，工业过程控制领域仍有近 90% 的回路在应用 PID 控制策略。PID 控制中一个关键的问题便是 PID 参数的整定，传统的方法是在获取对象数学模型的基础上，根据某一整定原则来确定参数。但是在实际的应用中，许多被控过程机理复杂，具有高度非线性、时变不确定性和纯滞后等特点。在噪声、负载扰动等因素的影响下，过程参数甚至模型结构均会随时间和工作环境的变化而变化。这就要求在 PID 控制中，不仅 PID 参数的整定不依赖于对象数学模型，并且 PID 参数能够在线调整，以满足实时控制的要求。自适应控制(Adaptive Control)是一门新兴的理论和技术，它是传统控制发展的高级阶段，主要用于解决那些传统方法难以解决的控制对象参数在大范围变化的问题，它的思路也就是解决 PID 参数在线调整问题的有效途径。

所谓自适应 PID 控制就是将自适应控制思想与常规 PID 控制器相结合，吸收了自适应控制与常规 PID 控制两者的优点。首先，它具备自适应能力，能够自动辨识被控过程参数、自动整定控制参数，能够适应被控过程参数的变化；其次，它又具有常规 PID 控制器结构简单、鲁棒性强、可靠性高、为现场工程设计人员所熟悉的优点。正是这两大优势，使得自适应 PID 控制成为过程控制的一种较理想的控制装置。

自适应 PID 控制器可分为两大类。一类基于被控过程参数(状态)辨识，统称为参数自适应 PID 控制器，其参数的设计依赖于被控过程模型参数的估计。另一类基于控制过程的某些特征参数，诸如临界振荡增益、临界振荡频率等。这种类型的自适应控制器被称为非参数自适应 PID 控制器。按照控制器参数设计原理来具体细分，前者可分为极点配置自适应 PID 控制器、相消原理自适应 PID 控制器、基于二次型性能指标的自适应 PID 控制器；后者包括神经网络自适应 PID 控制器、专家自适应 PID 控制器及模糊自适应 PID 控制器。下面分别予以简单介绍，然后我们重点描述模糊自适应 PID 控制器的设计与应用。

3.1 几种自适应 PID 控制

3.1.1 参数自适应 PID 控制

1. 极点配置自适应 PID 控制器

极点配置自适应控制算法是自适应控制中的一个重要组成部分，其主要思想是寻求一个反馈控制律，使得闭环传递函数的极点位于希望的位置。在此基础上提出的极点配置自适应 PID 控制算法原理就是通过调整 PID 参数，使系统具有期望的闭环特征方程。Tjokro 等人又针对具有未知或时变的纯滞后时间和可以测量的干扰噪声的被控对象，提出了极点配置自适应 PID 控制算法。其设计步骤是：①确定期望系统闭环极点位置；②在线估计、辨识系统参数；③计算控制器参数；④计算控制律。

极点配置自适应 PID 控制器具有计算量较小，鲁棒性较强，适合于非最小相位系统，且可方便地推广到多变量系统等优点。不足之处在于：其控制器的动态性能优劣依赖于极点配置的正确与否，而极点位置的配置又带有试凑性质。

2. 相消原理自适应 PID 控制器

利用控制器传递函数中的零极点对消被控对象传递函数的某些零、极点，从而使整个闭环系统工作在期望的状态上，这就是利用相消原理设计控制器的基本思想。为获得 PID 控制器的结构，利用这种原理设计控制器时，要求被控系统必须是二阶加纯滞后系统。Wittenmark 和 Åström 首先给出了基于相消原理的参数自适应 PID 控制算法，以后有了进一步的发展，提出了给定相位裕度和任意幅值裕度的自校正 PID 控制算法，并将算法在实践中进行了验证。用相消原理设计自适应控制器，具有原理简单、计算工作量小、易于工程实现等优点。但是该方法对于被控过程模型有较强的限制，因而不能应用于需要复杂控制和高性能要求的控制对象。

3. 基于二次型性能指标的 PID 控制器

欲获得参数优化的自适应 PID 控制器，最通用的方法是极小化某一个二次型指标函数。这种方法比较正规、系统，理论性较强，对不同的性能指标函数有不同的参数最优解，因此一直受到人们的重视。Song 等人在 1983 年提出了一种鲁棒性自适应反馈控制器的设计方法，这种控制器在结构和数学上均等价于常规的 PID 控制器。其算法的最大特点在于用鲁棒性概念设计控制器，并能保证在有界噪声和不可量测扰动存在的情况下控制误差是稳定和收敛的。此后又有人提出了参数逐步优化算法的自适应 PID 控制算法，能使 PID 控制器参数在线逐步优化，从而使这个系统的动态性能渐进最优化。

3.1.2　非参数自适应 PID 控制器

虽然参数自适应 PID 控制器是人们近年来研究的主要对象，但是人们并没有忽视对非参数自适应 PID 控制的研究。究其主要原因，就是非参数自适应 PID 控制器的设计不需要在线辨识被控对象的模型参数，避免了许多麻烦。自适应控制中的在线辨识占了计算机的大部分工作时间，而且存在闭环可辨识性问题。因此，对于高阶系统而言，不宜用参数自适应 PID 控制方法进行控制。但是非参数自适应 PID 控制就不同了，它只辨识某些特征值，因而它的一个最大优点是不受被控对象模型阶数的制约，计算机在线计算工作量小，实时性较强。

1. 神经网络自适应 PID 控制器

(1)PID 继电自整定与神经网络相结合的自适应 PID 控制系统

如图 3－1 所示，PID 继电自整定与神经网络相结合，共同完成 PID 自适应控制任务。以神经网络构造 PID 控制器，解决了 PID 参数在线调整的问题，使 PID 控制器适用范围更广。以继电自整定 PID 参数确定网络权的初值，使过程响应超调量降低，上升时间减短，控制质量提高。实施控制时，用开关 S 切换 PID 两种工作状态：参数整定和自适应控制。

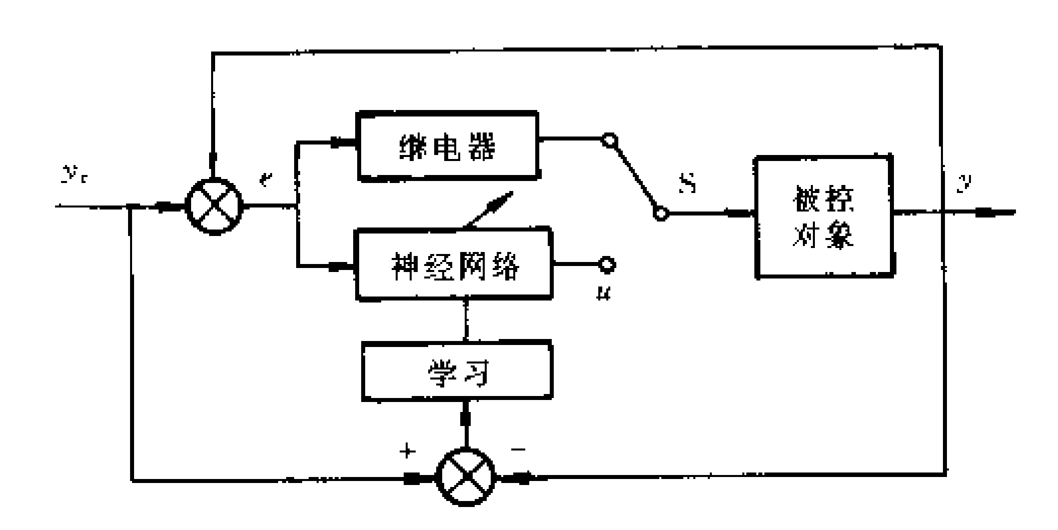


图 3－1　PID 继电自整定与神经网络相结合的 PID 自适应控制

(2)单神经元自适应 PID 控制器

用单神经元实现自适应 PID 控制的结构框图如图 3－2 所示。图中转换器的输入为设定值 y_r 及输出 y，转换器的输出为神经元学习控制所需要的状态量 x_1, x_2, x_3。神经元 PID 控制器的输出为

$$u(k) = u(k-1) + K\sum_{i=1}^{3} W_i(k)x_i(k)$$

式中，K 为神经元比例系数。

在单神经元控制器中引入输出误差平方的二次型性能指标，通过修改神经元控制器的加权系数 W_i，使性能指标趋于最小，从而实现自适应 PID 的最优控制。利用具有自学习和自适应能力的单神经元来构成单神经元自适应 PID 控制器，不但结构简单，学习算法物理意义明

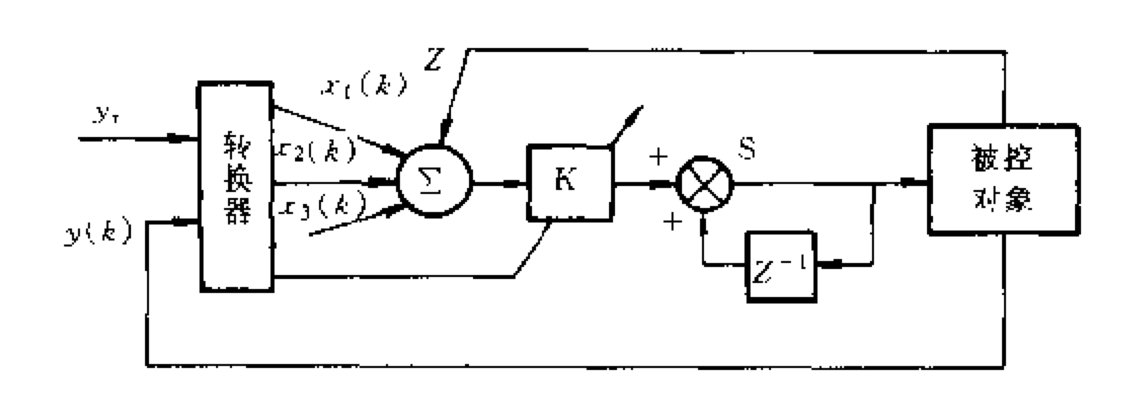


图 3-2　单神经元自适应 PID 控制器结构图

确，计算量小，且能适应环境变化，具有较强的鲁棒性。

2. 专家自适应 PID 控制器

具有专家系统的自适应 PID 控制器结构图如图 3-3 所示。它由参考模型、可调系统和专家系统组成。从原理上看，它是一种模型参考自适应控制系统。其中，参考模型由模型控制器和参考模型被控对象组成。可调系统由数字式 PID 控制器和实际被控对象组成。控制器的 PID 参数可以任意加以调整，当被控对象因环境原因而特性有所改变时，在原有控制器参数作用下，可调系统输出 $y(t)$ 的响应波形将偏离理想的动态特性。这时，利用专家系统以一定的规律调整控制器的 PID 参数，使 $y(t)$ 的动态特性恢复到理想状态。

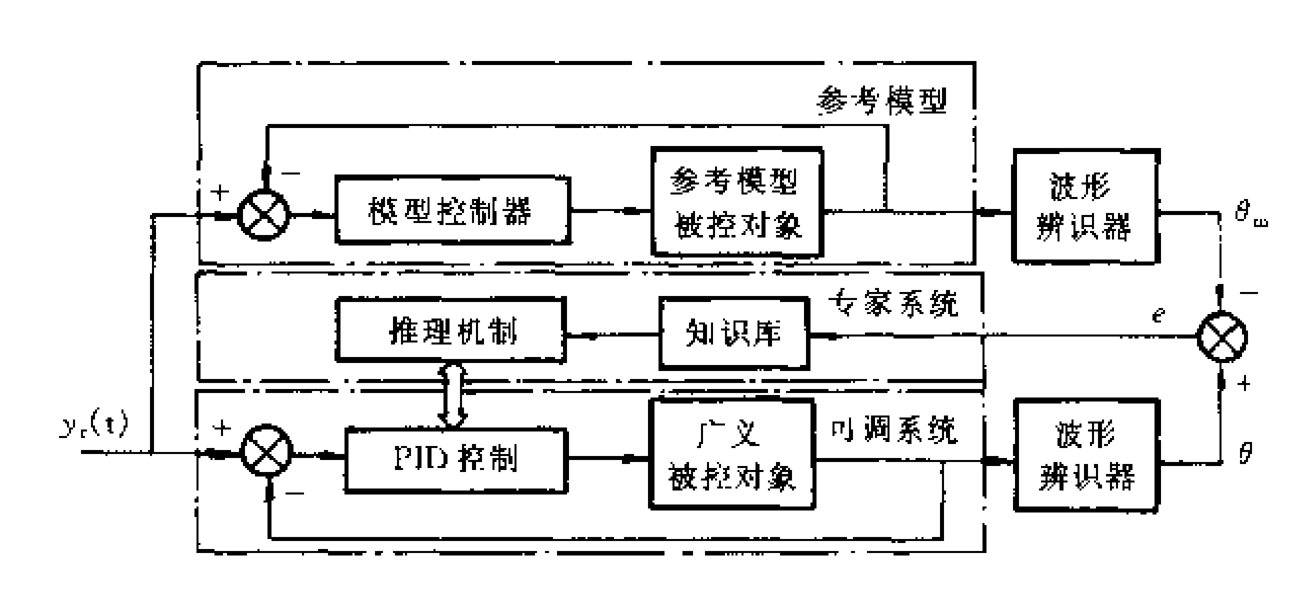


图 3-3　专家自适应 PID 控制原理图

专家系统由知识库和推理机制两部分组成，它首先检测参考模型和可调系统输出波形特征参数差值即广义误差 e。PID 自整定的目标就是调整控制器 PID 参数矢量 θ_C，使 θ 值逐步趋近于 θ_m（即 e 值趋近于 0）。

该系统由于采用闭环输出波形的模式识别方法来辨别被控对象的动态特性，不必加持续的激励信号，因而对系统造成的干扰小。另外，采用参考模型自适应原理，使得自整定过程可以根据参考模型输出波形特征值的差值来调整 PID 参数，这个过程物理概念清楚，并且避免

了被控对象动态特性计算错误而带来的偏差。

3. 模糊自适应PID控制器

模糊自适应PID(FAPID)控制系统结构图如图3-4所示。图中，FAC为模糊自适应控制器，与常规PID控制器一起组成FAPID控制器。FAPID控制器的设计分为独立的两步进行，简单方便。FAC的输出即为PID控制器的输入。PID参数若采用工程方法整定，可不需要被控对象模型。整定PID参数时，去掉FAC的作用。当在每个采样时刻获得了系统响应后，就可以根据此时刻系统响应偏离给定的情况及变化趋势，依据我们已有的系统控制知识，运用模糊控制方法，适当加大或减小控制力度，以控制响应朝偏离给定的方向变化，使输出尽快趋于稳定，可基于这种思路来设计FAC。模型规则物理意义明确，实时计算工作量小，便于工程应用。采用FAPID控制时，PID控制器中微分部分没有必要加入，因FAC已隐含微分作用。与传统PID控制比较，FAPID控制大大提高了系统的鲁棒性，减小了超调量，提高了抗干扰能力，缩短了调节时间。

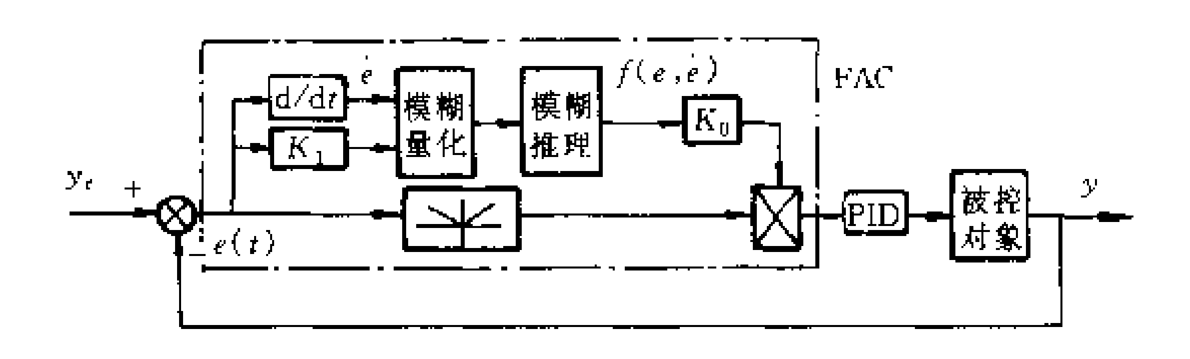


图3-4　FAPID控制系统结构图

3.2　两种类型的模糊PID控制器

3.2.1　调整系统控制量的模糊PID控制器

这类控制器的特点是在大偏差范围内利用模糊推理的方法调整系统的控制量U，而在小偏差范围内转换成PID控制，两者的转换根据事先给定的偏差范围自动实现。其控制结构图如图3-5所示。

下面以一个例子说明调整系统控制量的模糊PID控制器。

1. 控制器设计

(1) 模糊集及论域定义

对误差E、误差变化EC及控制量U的模糊集及其论域定义如下：

EC和U的模糊集均为

{NB,NM,NS,ZO,PS,PM,PB}

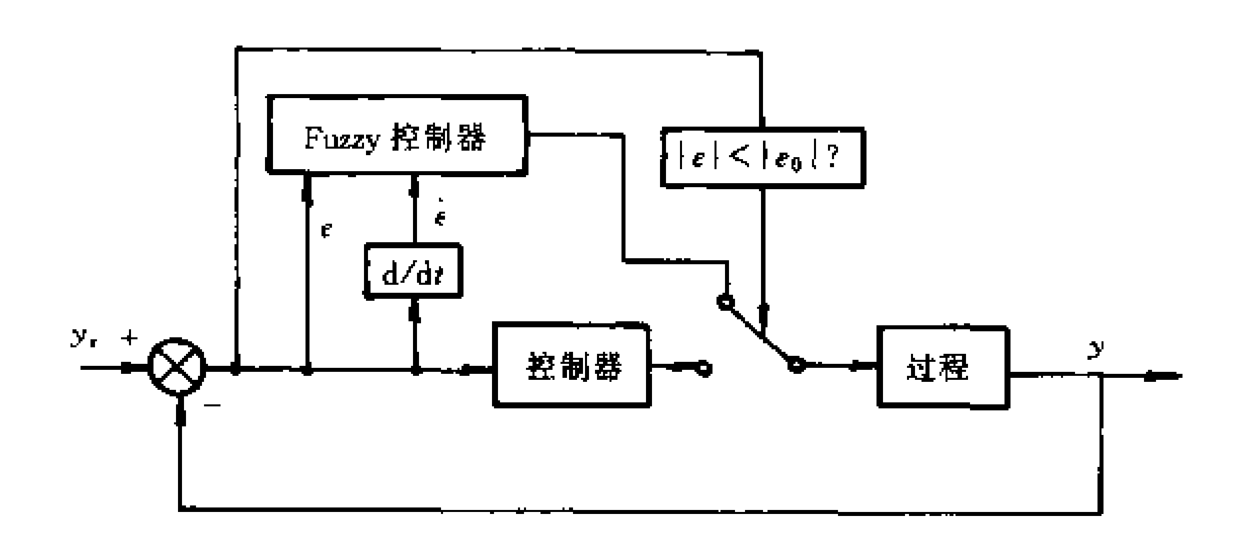


图 3-5　调整系统控制量的模糊 PID 控制

E 的模糊集为

{NB,NM,NS,NO,PO,PS,PM,PB}

E 和 EC 的论域均为

{−6,−5,−4,−3,−2,−1,0,1,2,3,4,5,6}

U 的论域为

{−7,−6,−5,−4,−3,−2,−1,0,1,2,3,4,5,6,7}

上述的误差模糊集选取 8 个元素,区分了 NO 和 PO,主要是着眼于提高稳态精度。E,EC 和 U 的隶属函数图形如图 3-6、图 3-7、图 3-8 所示。

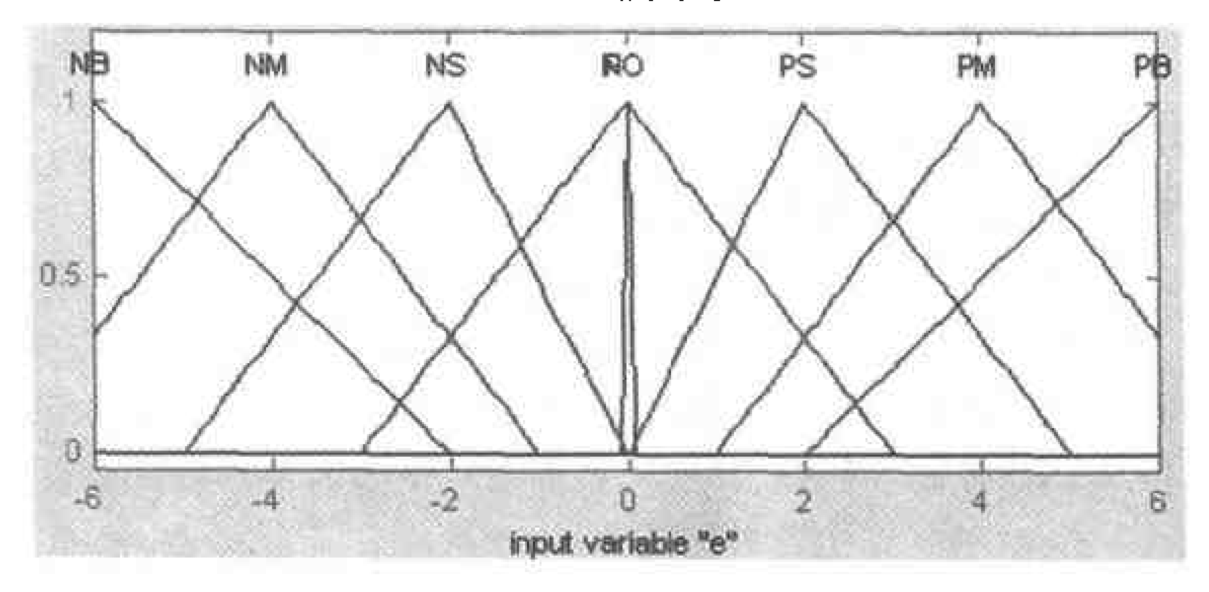


图 3-6　变量 E 的隶属函数

(2) 模糊控制规则

模糊控制规则如表 3-1 所示。

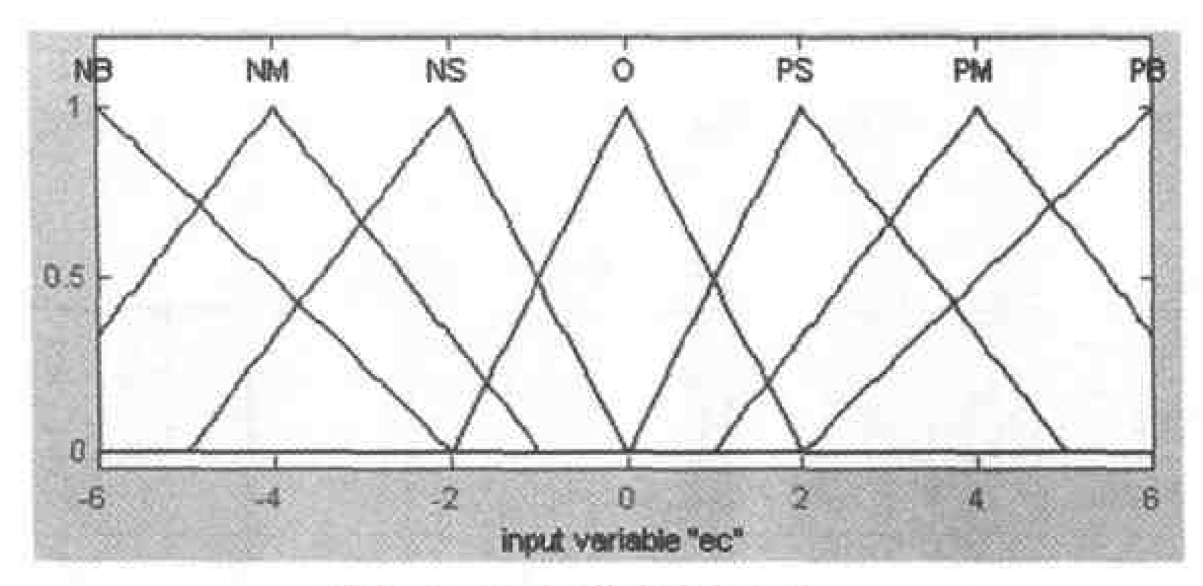


图3-7　变量 EC 的隶属函数

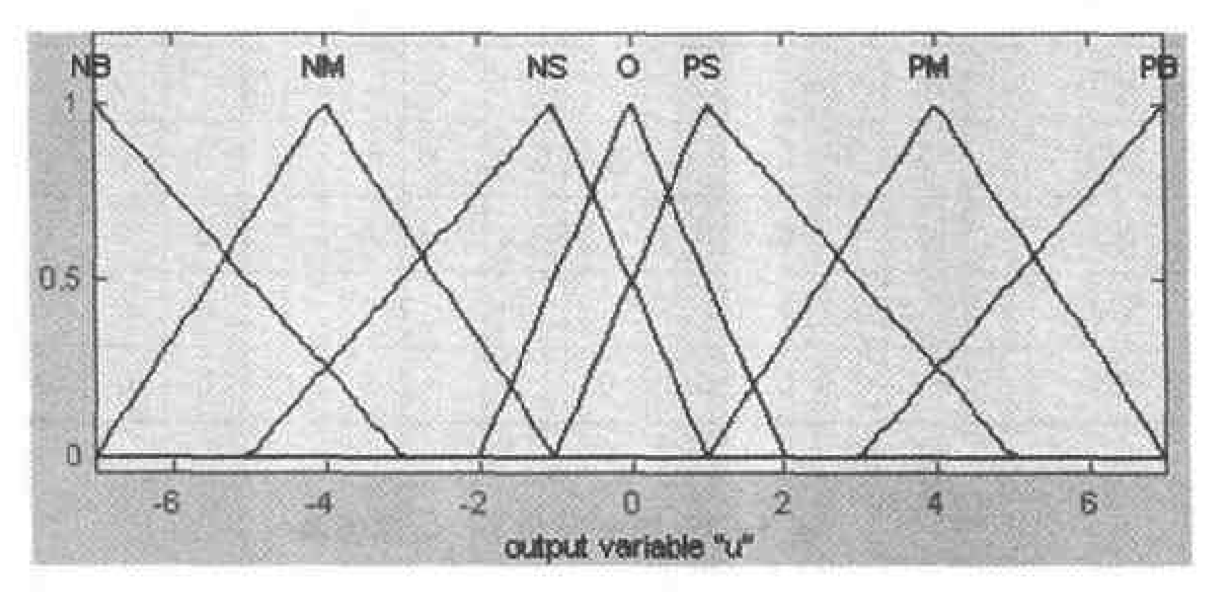


图3-8　变量 U 的隶属函数

表3-1　模糊控制规则

EC / *U* / *E*	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PB	PB	PM	ZO	ZO
NM	PB	PB	PB	PB	PM	ZO	ZO
NS	PM	PM	PM	PM	ZO	NS	NS
NO	PM	PM	PS	ZO	NS	NM	NM
PO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NM	NM	NM	NM
PM	ZO	ZO	NM	NB	NB	NB	NB
PB	ZO	ZO	NM	NB	NB	NB	NB

(3) 模糊变量的赋值表

模糊变量 E、EC 及 U 的赋值分别如表 3-2～表 3-4 所示。它们是根据不同对象的实际情况具体确定的。

表 3-2　模糊变量 E 的赋值表

E / U / E	-6	-5	-4	-3	-2	-1	-0	+0	+1	+2	+3	+4	+5	+6
PB	0	0	0	0	0	0	0	0	0	0	0.1	0.4	0.8	1.0
PM	0	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2
PS	0	0	0	0	0	0	0	0.3	0.8	1.0	0.5	0.1	0	0
PO	0	0	0	0	0	0	0	1.0	0.6	0.1	0	0	0	0
NO	0	0	0	0	0.1	0.6	1.0	0	0	0	0	0	0	0
NS	0	0	0.1	0.5	1.0	0.8	0.3	0	0	0	0	0	0	0
NM	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0	0
NB	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0	0

表 3-3　模糊变量 EC 的赋值表

EC / U / EC	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	+6
PB	0	0	0	0	0	0	0	0	0	0.1	0.4	0.8	1.0
PM	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2
PS	0	0	0	0	0	0	0	0.9	1.0	0.7	0.2	0	0
ZO	0	0	0	0	0	0.5	1.0	0.5	0	0	0	0	0
NS	0	0	0.2	0.7	1.0	0.9	0	0	0	0	0	0	0
NM	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0
NB	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0

表 3 - 4　模糊变量 U 的赋值表

U	−7	−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6	+7
PB	0	0	0	0	0	0	0	0	0	0	0	0.1	0.4	0.8	1.0
PM	0	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2	0
PS	0	0	0	0	0	0	0	0.4	1.0	0.8	0.4	0.1	0	0	0
ZO	0	0	0	0	0	0	0.5	1.0	0.5	0	0	0	0	0	0
NS	0	0	0	0.1	0.4	0.8	1.0	0.4	0	0	0	0	0	0	0
NM	0	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0	0
NB	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0	0	0

得到的模糊控制器的输出曲面如图 3 - 9 所示。

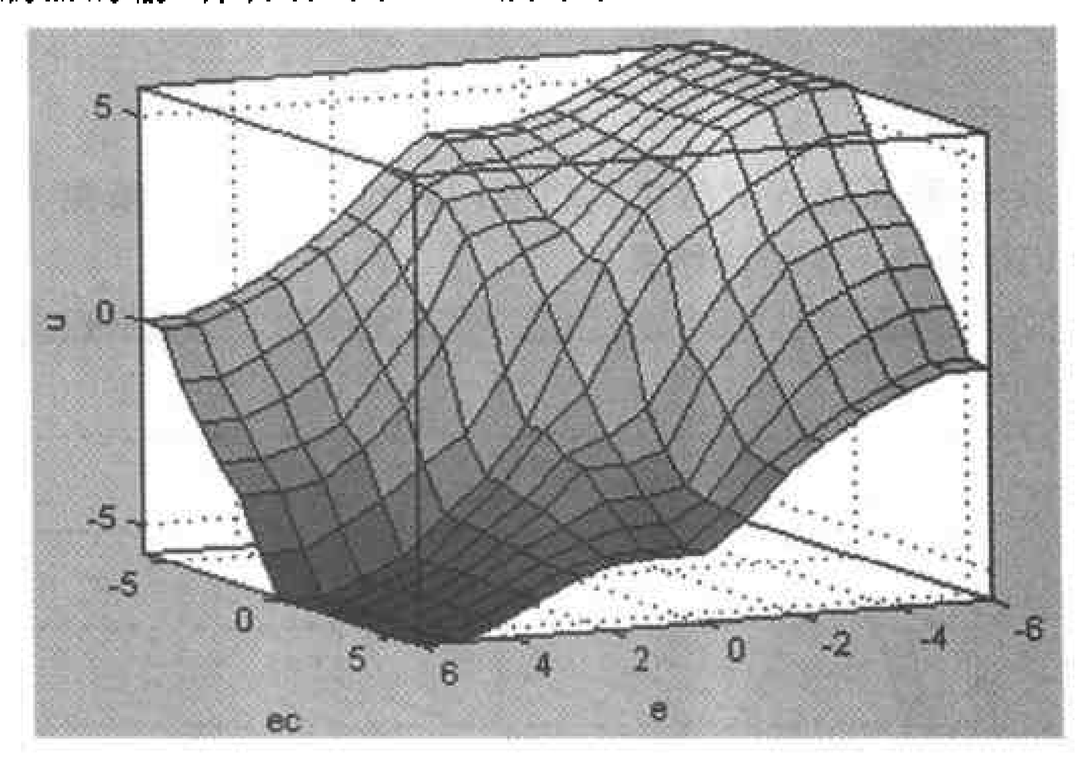


图 3 - 9　模糊推理输出曲面

2. 仿真模型的建立

根据以上的方法，利用 MATLAB 的 Fuzzy 工具箱及 Simulink 仿真环境对以下系统进行仿真(关于在 Simulink 仿真环境中建立模糊控制系统的方法见第 4 章)：

$$G(s) = \frac{1}{s^2 + 2s + 1}$$

其 Simulink 仿真计算图如图 3 - 10 所示。

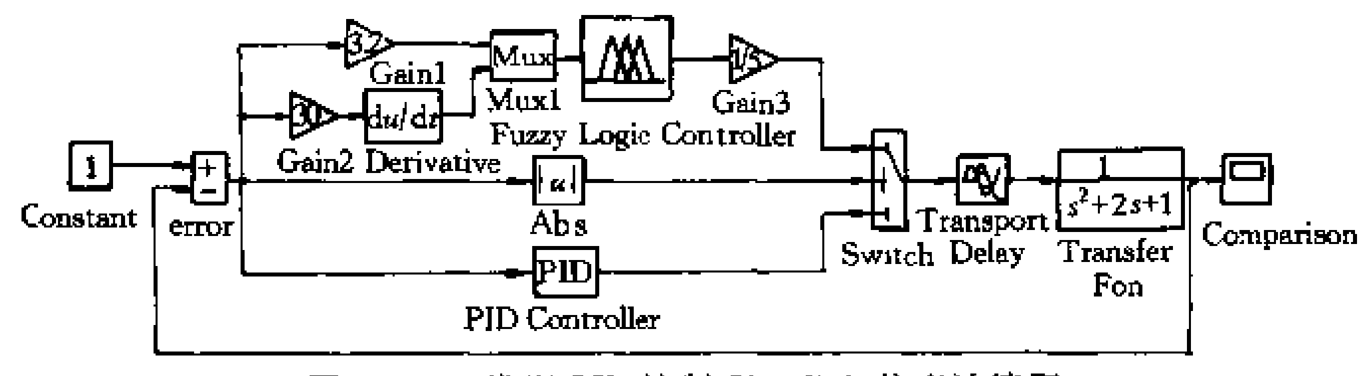


图 3－10　模糊 PID 控制 Simulink 仿真计算图

其中，PID 控制器的三个参数取值如下：

$$K_p = 0.85$$

$$K_i = 1.0$$

$$K_d = 0.5$$

其含义分别为该控制器的比例增益、积分增益和微分增益。

模糊控制和 PID 控制转换的设定值为：

$$|e_0| = 0.5$$

3. 仿真结果

图 3－11 为 $G(s)$ 在阶跃输入下的仿真结果，其中实线表示在模糊 PID 控制下的仿真结果，虚线表示常规 PID 控制下的仿真结果。

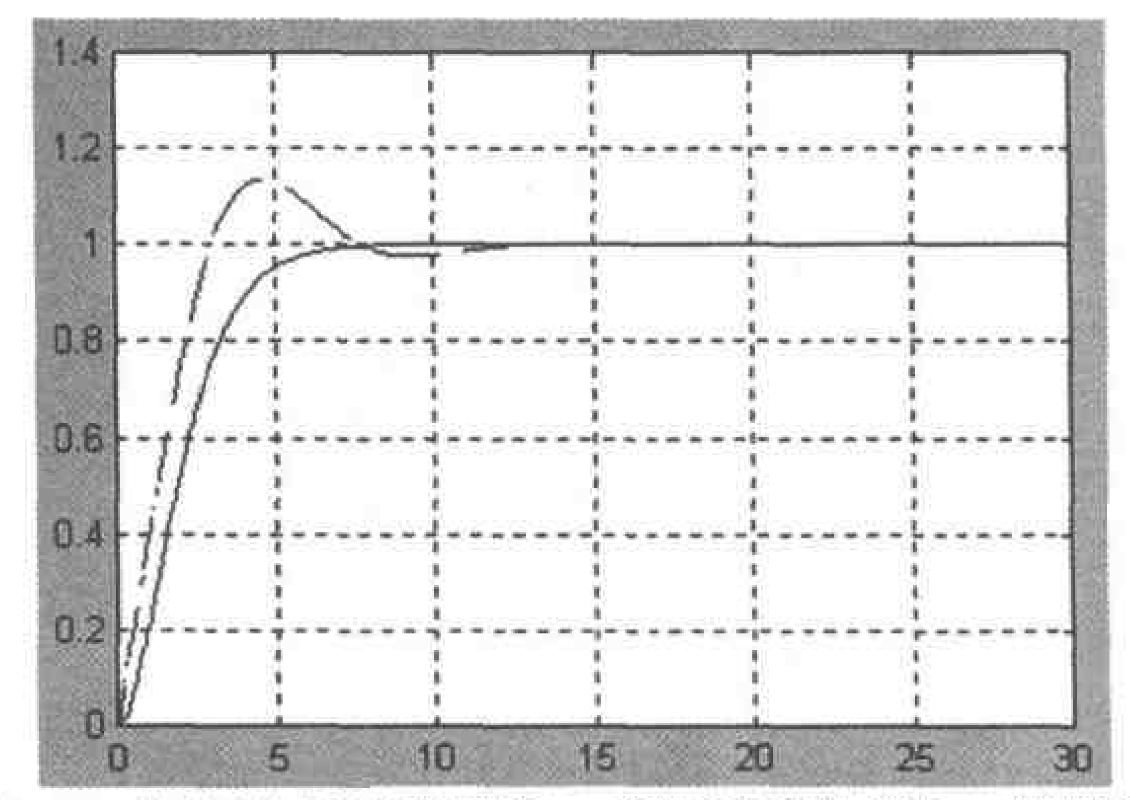


图 3－11　控制系统性能比较（虚线——常规 PID 控制，实线——模糊控制）

图 3－12 是加上了延迟环节以后，在两个不同控制器作用下的仿真结果，所加的延迟环节

为 $e^{-0.5s}$。

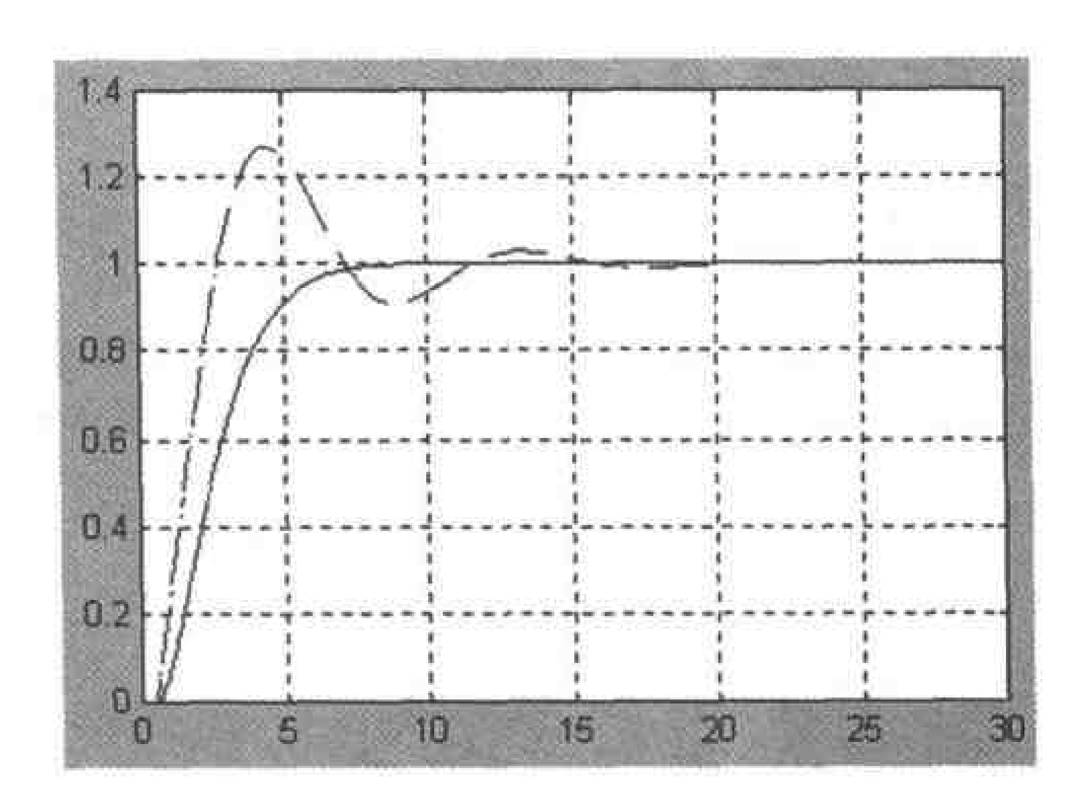


图 3-12　加延迟环节后系统性能比较(虚线——常规 PID 控制,实线——模糊控制)

4. 模糊控制量调整 PID 控制器结论

从仿真曲线和性能指标我们可以看出,与常规 PID 控制相比,模糊 PID 控制器能使系统响应的超调减小,反应时间加快。尤其是在系统具有延迟及模型结构和参数不确定的情况下,模糊 PID 控制具有更佳的控制效果。

3.2.2　模糊增益调整 PID 控制器

模糊增益调整 PID 控制器就是利用模糊规则和推理来对 PID 参数进行调整,其结构图如图 3-13 所示。

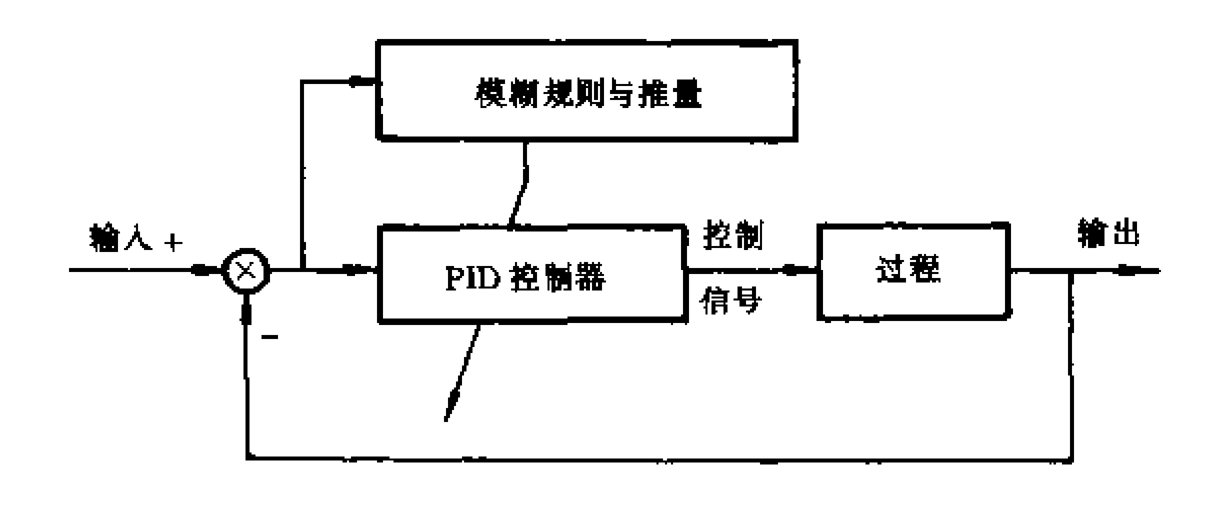


图 3-13　具有模糊增益调整的 PID 控制器

假定 K_p、K_i，K_d 分别表示比例增益、积分增益和微分增益。为了方便，可将其归一化为 0～1 间的参数 K_p 和 K_d。K_i 可表示为

$$K_i = K_p'^2 / (\alpha K_d')$$

参数 K_p'，K_d' 和 α 由当前误差 $e(k)$ 和它的一阶差分 $\Delta e(k)$ 根据一组模糊规则决定：

If $e(k)$ 是 A_i 和 $\Delta e(k)$ 是 B_i，then K_p 是 C_i，K_d 是 D_i 和 $\alpha = \alpha_i (i = 1, 2, \ldots, m)$

这里 A_i，B_i，C_i 和 D_i 是在相应支集上的模糊集合；$e(k)$ 和 $\Delta e(k)$ 的隶属函数如图 3-14 所示。模糊集合 C_i 和 D_i 或小或大，如图 3-15 所示。α_i 是常数，其隶属函数如图 3-16 所示。

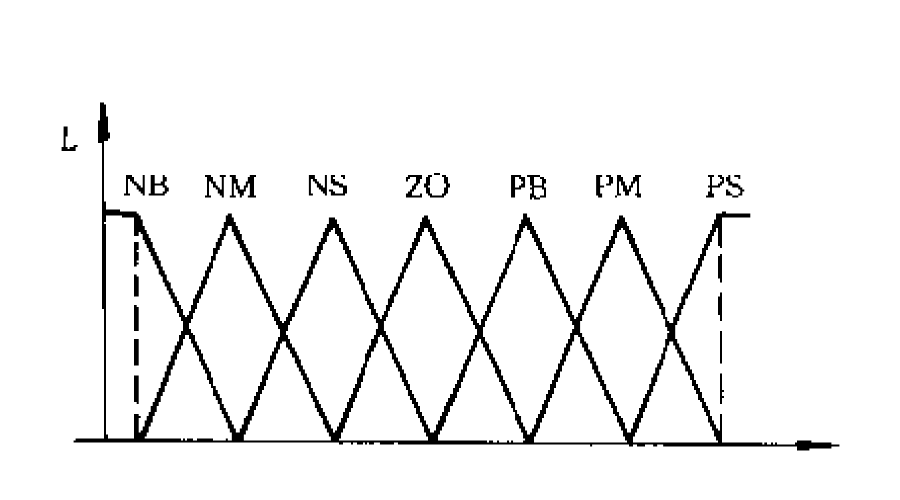


图 3-14　$e(k)$ 和 $\Delta e(k)$ 的隶属函数

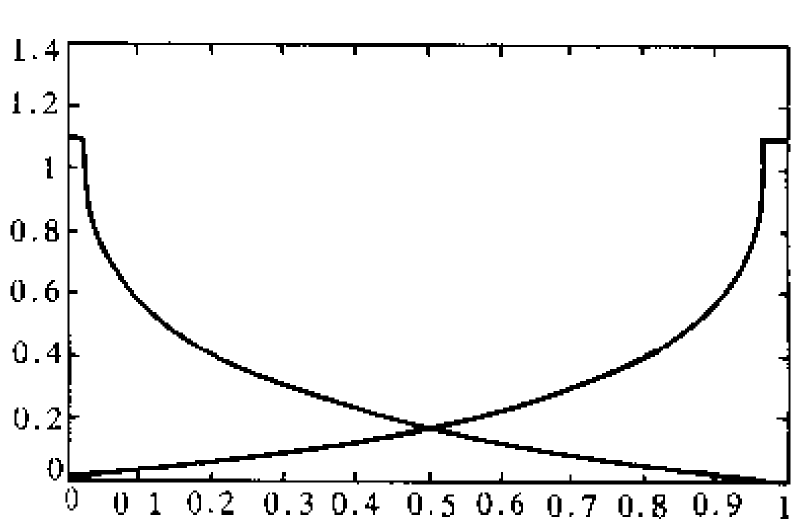


图 3-15　K_p 或 K_d 的隶属函数

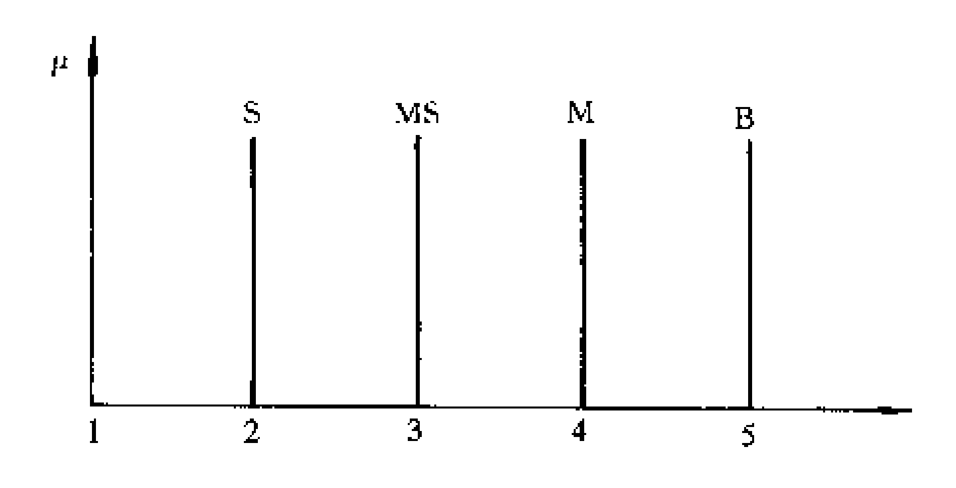


图 3-16　α 单点隶属函数

K_p'，K_d' 和 α 的模糊调整规则如表 3-5～表 3-7。

表 3-5　K'_p 模糊调整规则

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

表 3-6　K'_d 模糊调整规则

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZO	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

表 3-7　α 调整规则

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	2	2	2	2	2	2	2
	NM	3	3	2	2	2	3	3
	NS	4	3	3	2	3	3	4
	ZO	5	4	3	3	3	4	5
	PS	4	3	3	2	3	3	4
	PM	3	3	2	2	2	3	3
	PB	2	2	2	2	2	2	2

由表3－5～表3－7得到K_p'、K_d'和α，根据下式可求得PID的参数。

$$\left.\begin{aligned}K_p&=(K_{p,max}-K_{p,min})K_p'+K_{p,min}\\K_d&=(K_{d,max}-K_{d,min})K_d'+K_{d,min}\\K_i&=K_p^2/(\alpha K_d)\end{aligned}\right\}$$

K_p和K_d的范围如下：

$$K_{p,min}=0.32K_u$$
$$K_{p,max}=0.6\ K_u$$
$$K_{d,min}=0.08\ K_uT_u$$
$$K_{d,max}=0.15\ K_uT_u$$

K_u和T_u分别为比例控制下的增益和振荡周期。

1. 系统描述

根据以上的方法，对以下两个系统进行了仿真：

$$G_1(s)=\frac{4.228}{(s+0.5)(s^2+1.64s+8.456)}$$

$$G_2(s)=\frac{27}{(s+1)(s+3)^3}$$

其中，对于$G_1(s)$，K_u取4，T_u取2.9，对于$G_2(s)$，K_u取5，T_u取3.4。

得到的模糊控制器的输出曲面如图3－17所示。

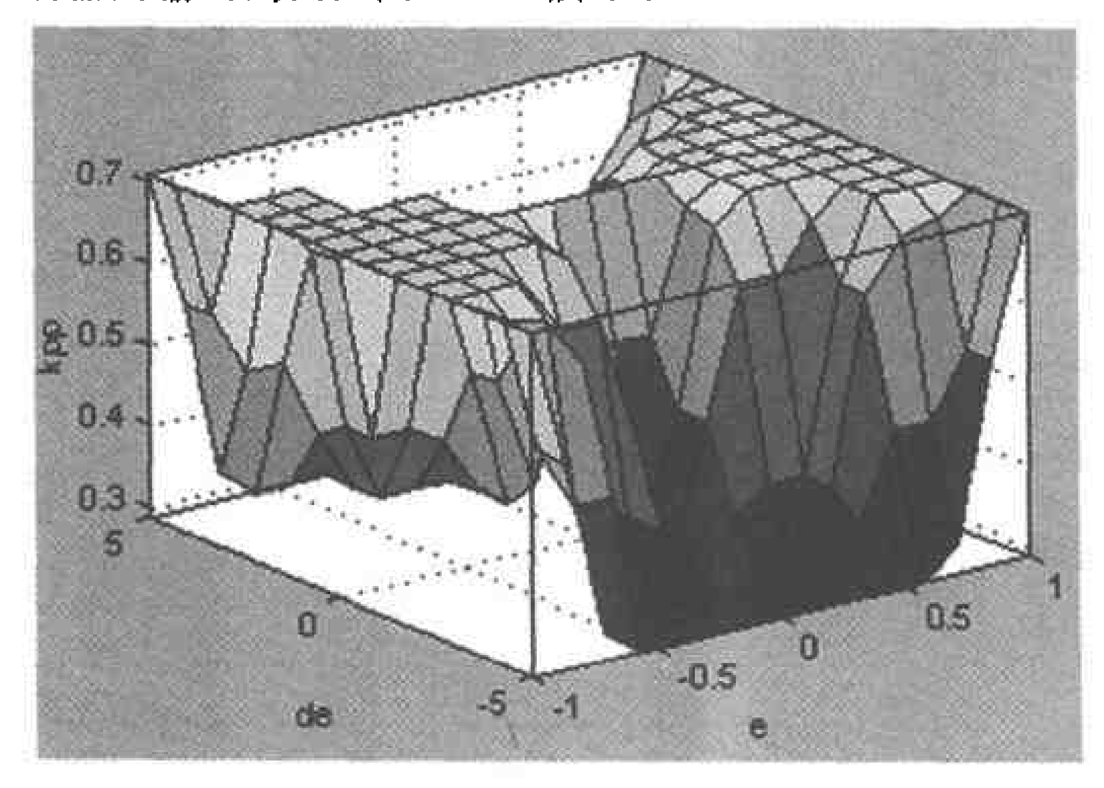


图3－17　模糊控制器的输出曲面

2. 建立仿真模型

根据系统描述，可建立该模糊控制器的仿真模型，如图3-18所示。

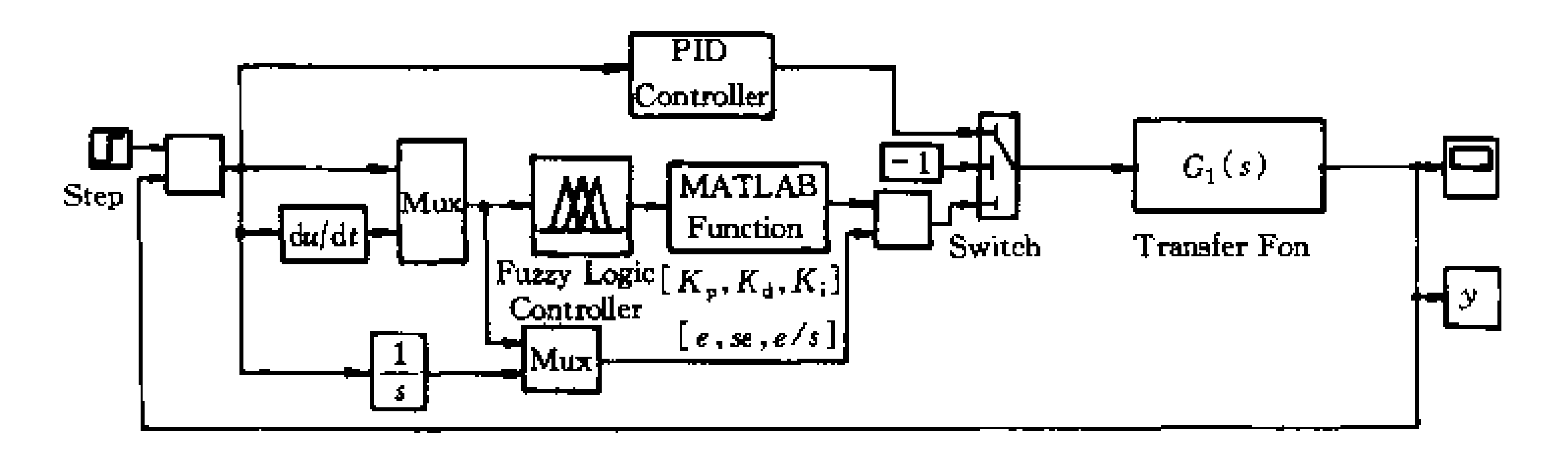


图3-18　模糊增益调整PID控制器Simulink仿真框图

注：MATLAB Function用于将归一化的参数转化为实际PID参数，即完成其工作。

3. 仿真结果

图3-19和图3-20分别为$G_1(s)$和$G_2(s)$在阶跃输入下的仿真结果，并与常规PID方法(Zeigler-Nichols方法)进行了比较。表3-8列出了模糊控制与常规PID控制的性能指标。

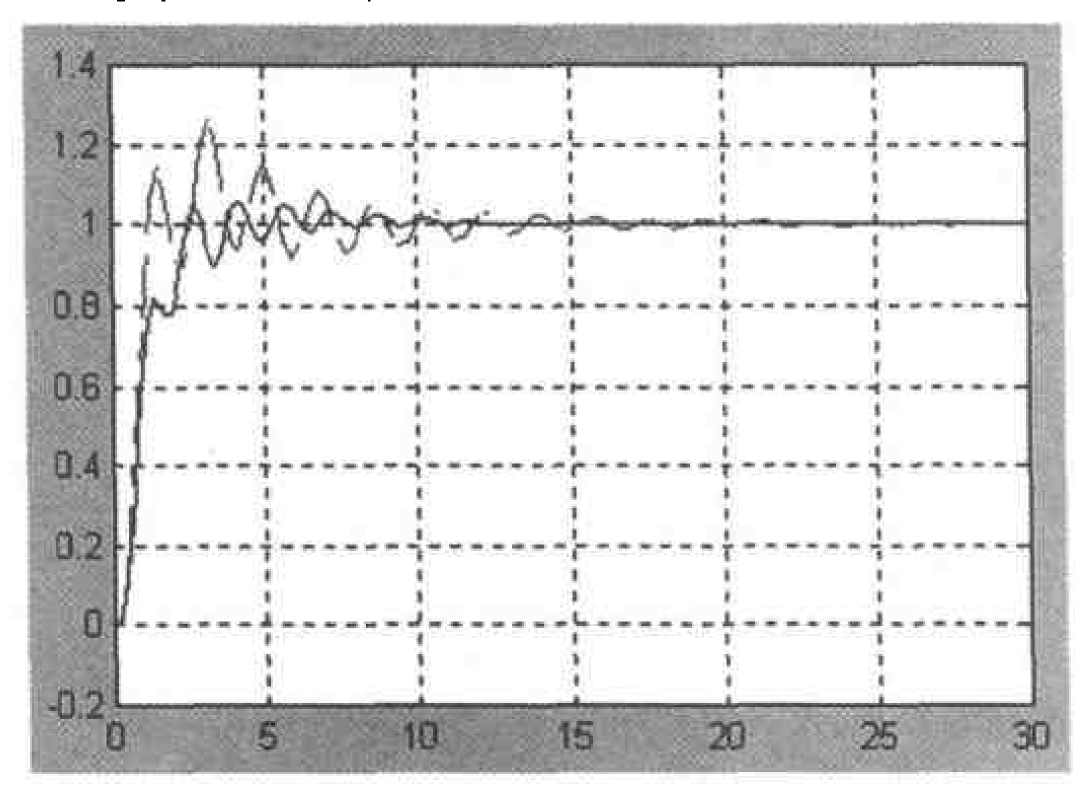


图3-19　三阶系统性能比较(虚线——常规PID控制，实线——模糊控制)

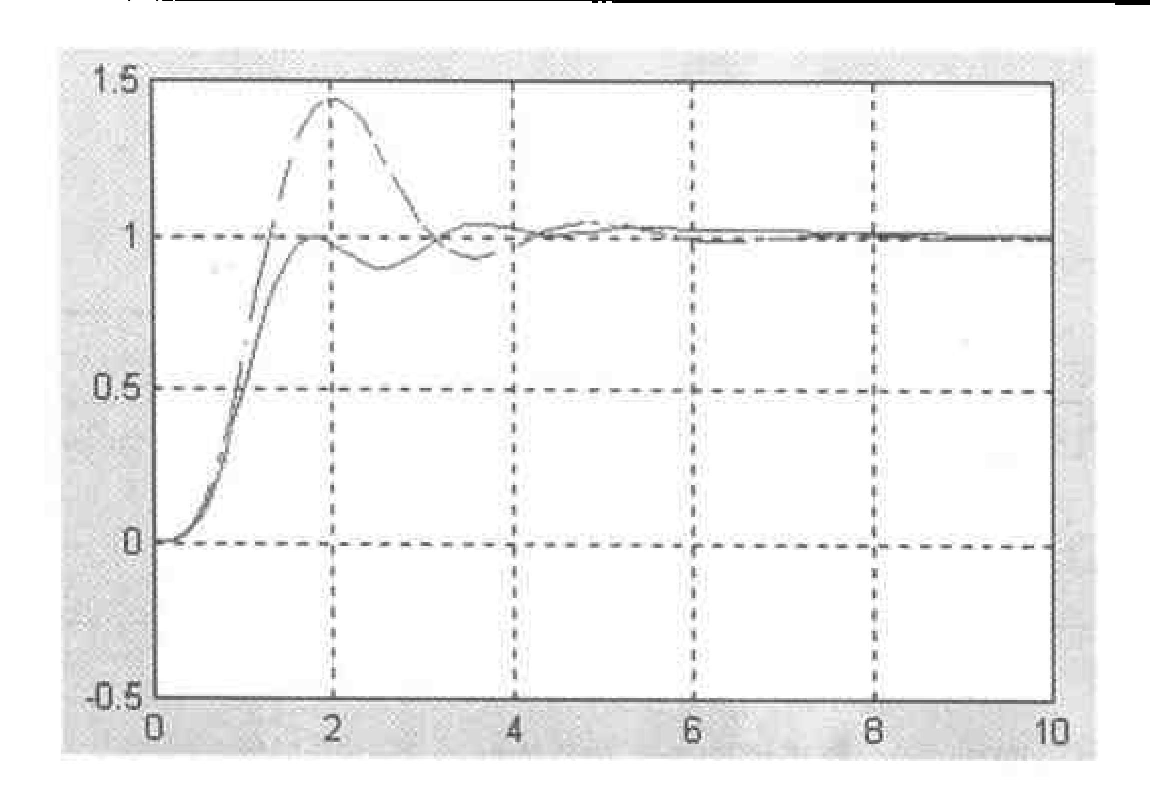


图 3-20 四阶系统性能比较(虚线——常规 PID 控制,实线——模糊控制)

表 3-8 性能指标

被控对象	常规 PID 控制		模糊控制
$G_1(s)$	$K_p=2.19$	$\sigma\%=20\%$ $t_s=5.76s$	$\sigma\%=8.6\%$ $t_s=4.78s$
	$T_i=1.03$		
	$T_d=0.258$		
$G_2(s)$	$K_p=3.072$	$\sigma\%=44.5\%$ $t_s=3.87s$	$\sigma\%=3.7\%$ $t_s=2.98s$
	$T_i=1.325$		
	$T_d=0.338$		

注:$\sigma\%$ 为超调量,t_s 为调节时间(取 5% 的条形域)

从仿真曲线和性能指标我们可以看出,与常规 PID 控制相比,具有模糊增益调整的 PID 控制器使系统响应的超调减小,反应时间加快。



模糊控制应用实例

模糊控制在各方面都得到了广泛的应用，利用 MATLAB 模糊逻辑工具箱进行模糊控制系统的设计和验证(仿真)非常方便。下面我们给出了一些工程技术人员以及作者在控制系统的设计和实践中的应用，对于某些内容，为了反映其完整的思路与工作流程，我们作了较大篇幅的收录。为了使读者轻松地掌握模糊逻辑工具箱的使用，在第一个应用实例中，我们详细说明了应用 MATLAB 模糊逻辑工具箱以及 Simulink 工具建立模糊控制器和动态仿真模型的具体步骤。由于在 MATLAB 的应用及其界面的显示中，不能将各变量及控制规律表示为科技文献的记录格式，因此下面给出的变量及控制规律一般都是在 MATLAB 中能够表示出来的记法。

4.1 模糊控制器与 Simulink 的连接

MATLAB 提供的 Simulink 是一个用来对动态系统进行建模、仿真和分析的软件包。它使用简单，功能强大，并且支持连续、离散及两者混合的线性和非线性系统。Simulink 包含多个子模型库，每个子模型库中又包含多个功能模块。利用鼠标可直观地“画”出系统模型，然后直接进行仿真。仿真时可选择各种数值算法、仿真步长等重要参数，并可用模拟示波器将仿真动态结果予以显示，直观高效。

Simulink 可以与模糊逻辑工具箱实现完美的结合。关于 Simulink 的使用，请参见相关的介绍，在此只介绍如何将已经建立的模糊控制器或模糊控制系统嵌入 Simulink 仿真中。而模糊逻辑工具箱最大的优点之一，也同样是它可以直接将模糊系统嵌入 Simulink，并在该仿真环境中测试系统输出。

将模糊系统嵌入 Simulink 的步骤如下。

1. 第 1 步：载入模糊推理系统

要将模糊系统嵌入 Simulink，首先应保证与模糊系统相应的模糊推理系统(FIS)结构已同时装载在 MATLAB 工作区中(而不是存在磁盘中)，并由相关的名字指向模糊逻辑控制器。

例如，假定已经建立了一个名为“aaa”的模糊控制器或模糊控制系统，则单击模糊逻辑工具箱 5 个 GUI 界面中任一个的 File 菜单，选择其中的“Save to workspace”，将其保存到 MATLAB 的工作区中；如果已建立的模糊控制器或模糊控制系统尚未命名，则选择“Save to workspace as...”，以“aaa”为模糊控制系统的名称将其存在 MATLAB 工作区，如图 4－1 所示。

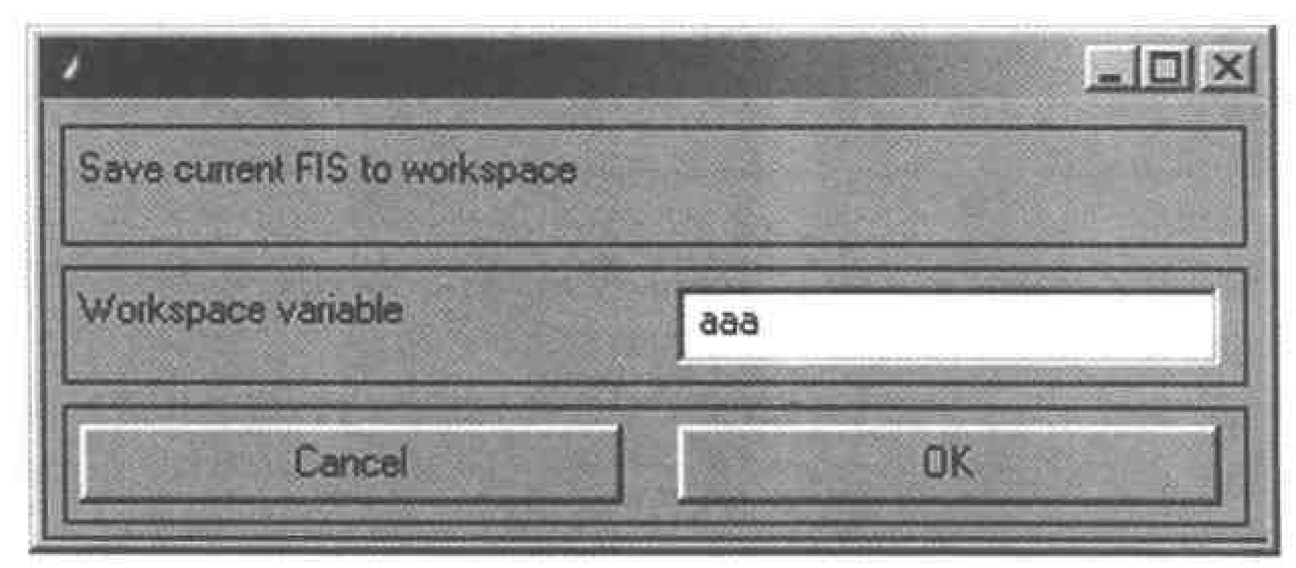


图 4－1 将模糊推理系统保存到工作区

2. 第 2 步：打开 Simulink 模型

在 MATLAB 工作区内用命令“Simulink”建立或打开自己的 Simulink 仿真模型。由于采用的是模糊控制方法，因此需要在打开的 Simulink 库（Simulink Library Browser）中，选择“Fuzzy Logic Toolbox”项。其下有两个选择对象，分别为“Fuzzy Logic Controller”和“Fuzzy Logic Controller with Ruleviewer”。两者的图标如图 4－2 所示。

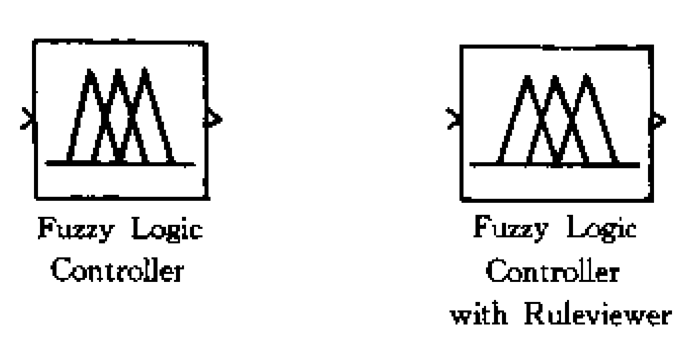


图 4－2 Simulink 中的模糊逻辑控制器模块图标

两者的功能完全一样。不同之处仅在如果选择后者，则在 Simulink 进行仿真过程时，将打开模糊规则观察器，从观察器上可以看到的只是已建立的模糊控制器或模糊控制系统的模糊规则而已。

将选项“Fuzzy Logic Controller”或“Fuzzy Logic Controller with Ruleviewer”拖到 Simulink 仿真系统中，双击其图标，出现如图 4－3 所示的对话框。

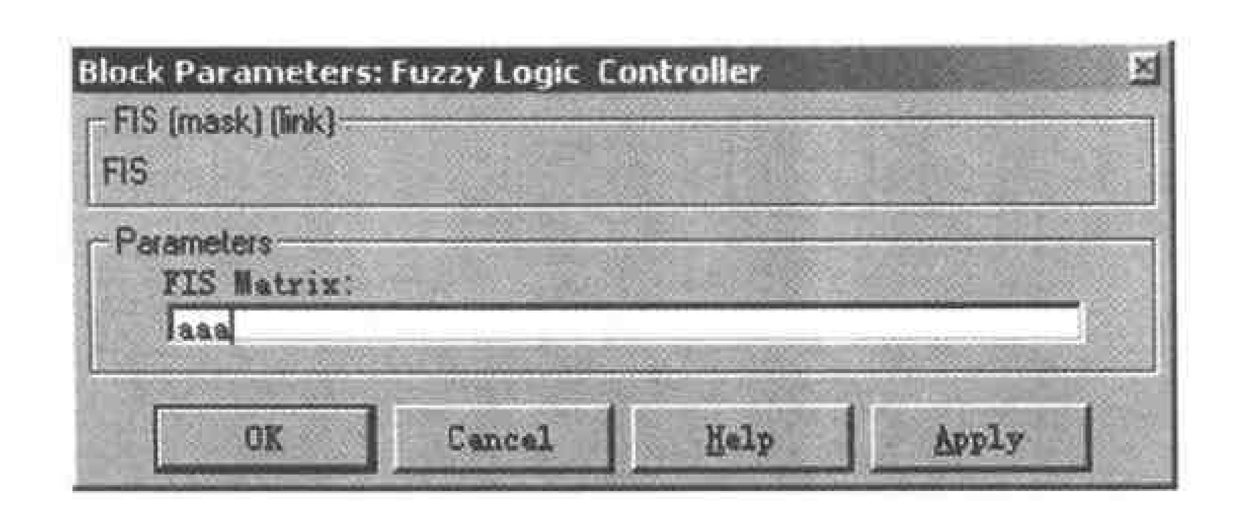


图 4-3 连接模糊控制器与模糊推理系统对话框

在 FIS Matrix 文本框中输入“aaa”并按“OK”按钮，则将工作区中的 FIS 结构与该模糊控制器连接起来。

3. 第 3 步：设定模糊控制器的输入与输出

一个 FIS 结构的输入变量的个数往往多于一个，但因为“Fuzzy Logic Controller”的图标是单输入的，因此需要用到一个向量信号组合工具。单击 Simulink 库下 Simulink 项的加号将其展开，再将 Simulink 项下的 Signal & System 项展开，选择其中的 Mux 对象并拖到 Simulink 仿真系统中，其输出与模糊逻辑控制器相连，其输入则与 FIS 结构的输入变量相连。双击其图标，出现如图 4-4 所示的对话框。在此对话框中可以确定输入的个数。

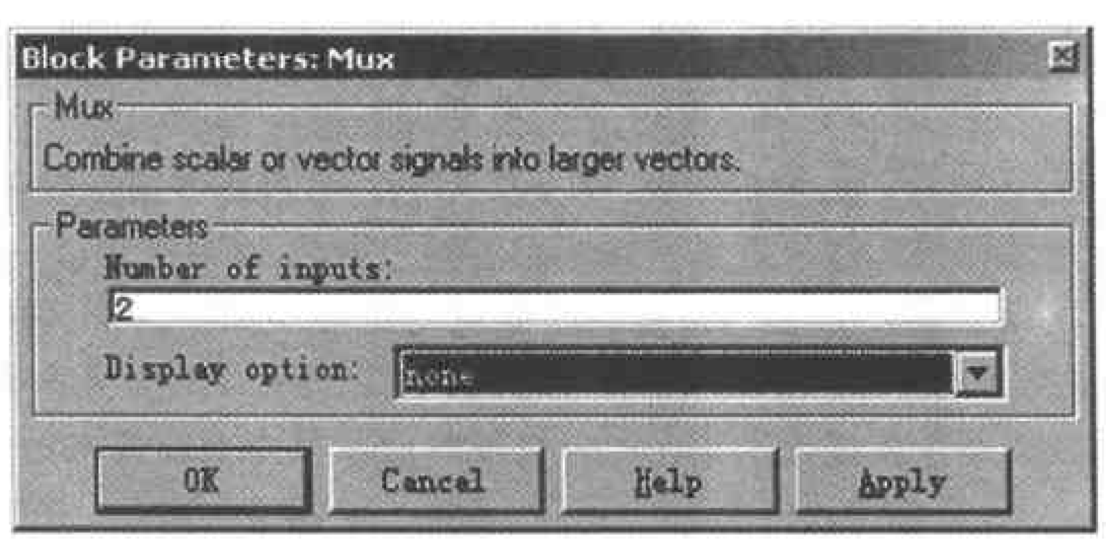


图 4-4 设定模糊控制器输入变量的数目

至此，则完成了将模糊系统嵌入 Simulink 仿真系统的工作。图 4-5 给出的是一个图例。但它仅是一个图例而已，不能完成任何工作。

需要注意的是，在应用任何一个编辑器来修改模糊推理系统之前，最好先停止仿真过程。并要记住的是，在重新启动仿真之前，要将对 FIS 的任何改变保存到工作区或磁盘中。

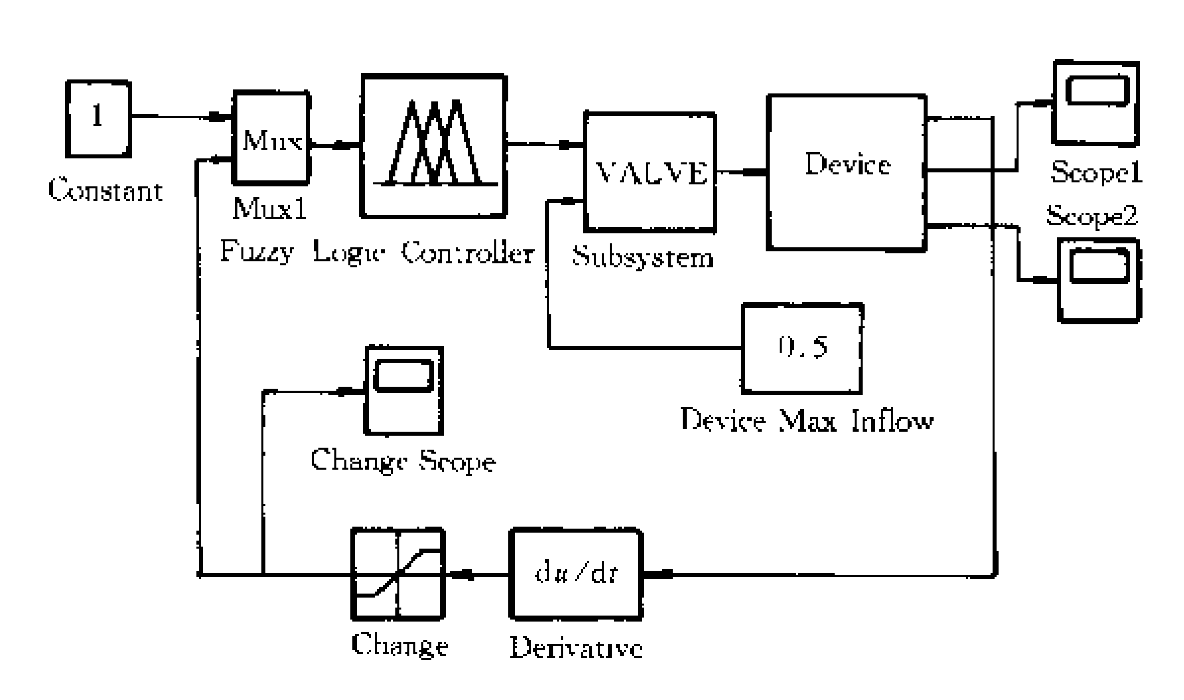


图 4-5　模糊控制仿真系统图例

4.2　模糊 PD 控制器

在经典控制器的设计中，常常将系统偏差和系统偏差的微分信号综合起来，构成 PD 调节器，其输出 U 作为被控对象的输入信号。加入系统偏差微分信号的目的，是赋予调节器以某种程度的预见性，以获得更良好的控制品质。

PD 调节器的控制规律是

$$u = K_e e - S_2 \frac{\mathrm{d}e}{\mathrm{d}t}$$

在 PD 调节器的设计中，较难精确确定的是其参数。而根据专家经验，采用模糊方法来设计，则可以比较容易地获得理想的效果。在下面的例子中，先对系统偏差及其微分信号进行模糊量化处理，得到模糊变量 E 和 EC，按模糊控制规则进行模糊决策得到模糊控制量 U，再经过反模糊和比例变换得到实际控制量输出。如图 4-6 所示。其中，判断系统偏差的大小、系统偏差的微分的大小(即系统偏差变化速度的快慢)、以及根据系统偏差的大小及其变化速度

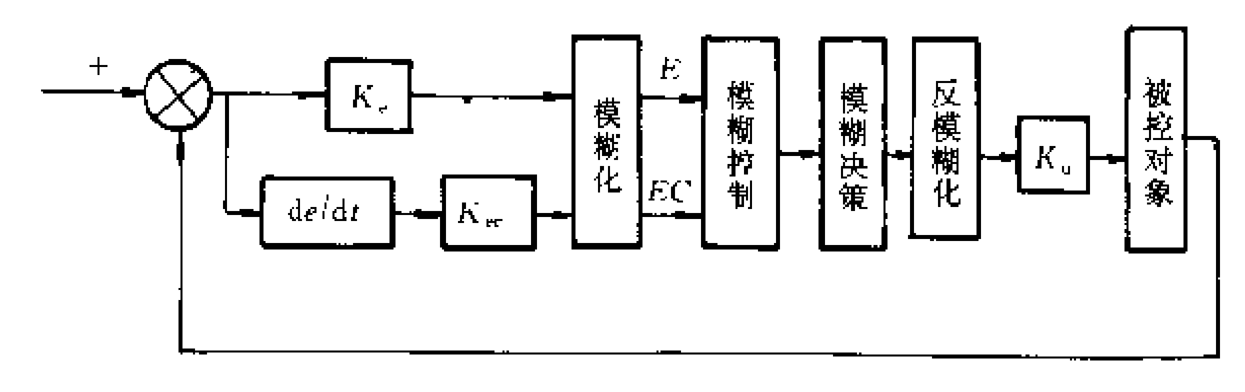


图 4-6　模糊控制结构

的快慢来确定模糊规则，均需要一定的工作经验并依据实际系统的特性进行，即前面所说的依照专家经验确定模糊集和模糊规则。而模糊推理和反模糊化通常有 Mamdani 方法和 Sugeno 方法两种，一般而言，它们的输出结果没有太大的不同。

4.2.1 用模糊逻辑工具箱设计模糊控制器

1. 模糊集与隶属函数的建立

若取 E、EC 的论域均为[−6,6]，其模糊子集都为{NB,NM,NS,ZO,PS,PM,PB}。共有49条模糊控制规则。U 的论域为[−1,1]，模糊子集为{NB,NM,NS,ZO,PS,PM,PB}。下面利用 MATLAB 模糊逻辑工具箱建立各变量的隶属函数。

在 MATLAB 中键入命令"fuzzy"，进入模糊逻辑编辑窗口"FIS Editor"。此时模糊逻辑工具箱默认建立的是一个单输入输出的 Mamdani 型推理系统。如果需要建立的是 Sugeno 型推理系统，请在 File 菜单下拉菜单中选择"New Sugeno FIS"。如果要打开一个已经建立好的模糊推理系统，请在 File 菜单下拉菜单中选择"Open FIS from Disk..."。

在此，我们采用默认的 Mamdani 型推理系统。由于要建立的是一个二输入单输出系统。因此需要添加一个输入变量。请在 Edit 菜单下拉菜单中选择"Add input"。此时 FIS 编辑器的 GUI 界面会更新为一个二输入单输出系统。其输出变量为"input1"和"input2"。输出变量为"output1"。

单击 FIS 编辑器 GUI 界面左上区域输入变量"input1"的图标，此时该图标将被一个红色方框环绕，表示 input1 成为当前变量。在 FIS 编辑器 GUI 界面右下区的"Current Variable"的"name"文本框中，将变量 input1 的名称改为"E"。

按同样的方法，将变量 input2 的名称改为"EC"，将变量 output1 的名称改为"U"。

在 FIS 编辑器 GUI 界面的左下区域，可以选择模糊交、模糊并、模糊推理、模糊决策和反模糊化方法（对于 Sugeno 型推理系统，模糊推理和模糊聚类是不可选的）。在此我们采用其默认值。

在 File 下拉菜单中选择"Save to disk"，将模糊推理系统以"myexample"为名称保存到磁盘中。

此时 FIS 编辑器的 GUI 界面显示如图 4-7 所示。其中在 GUI 界面的中部给出了模糊推理系统的名称和类型。

下面建立各变量的隶属函数。

在 FIS 编辑器的 GUI 界面上双击变量 E（或从 FIS 编辑器的 View 下拉菜单中选择"Edit membership functions..."），打开隶属函数编辑器。

与 FIS 编辑器一样，在隶属函数编辑器的左上区域，单击某一输入或输出变量的图标，可使该变量成为当前变量，从而编辑该变量的隶属函数。

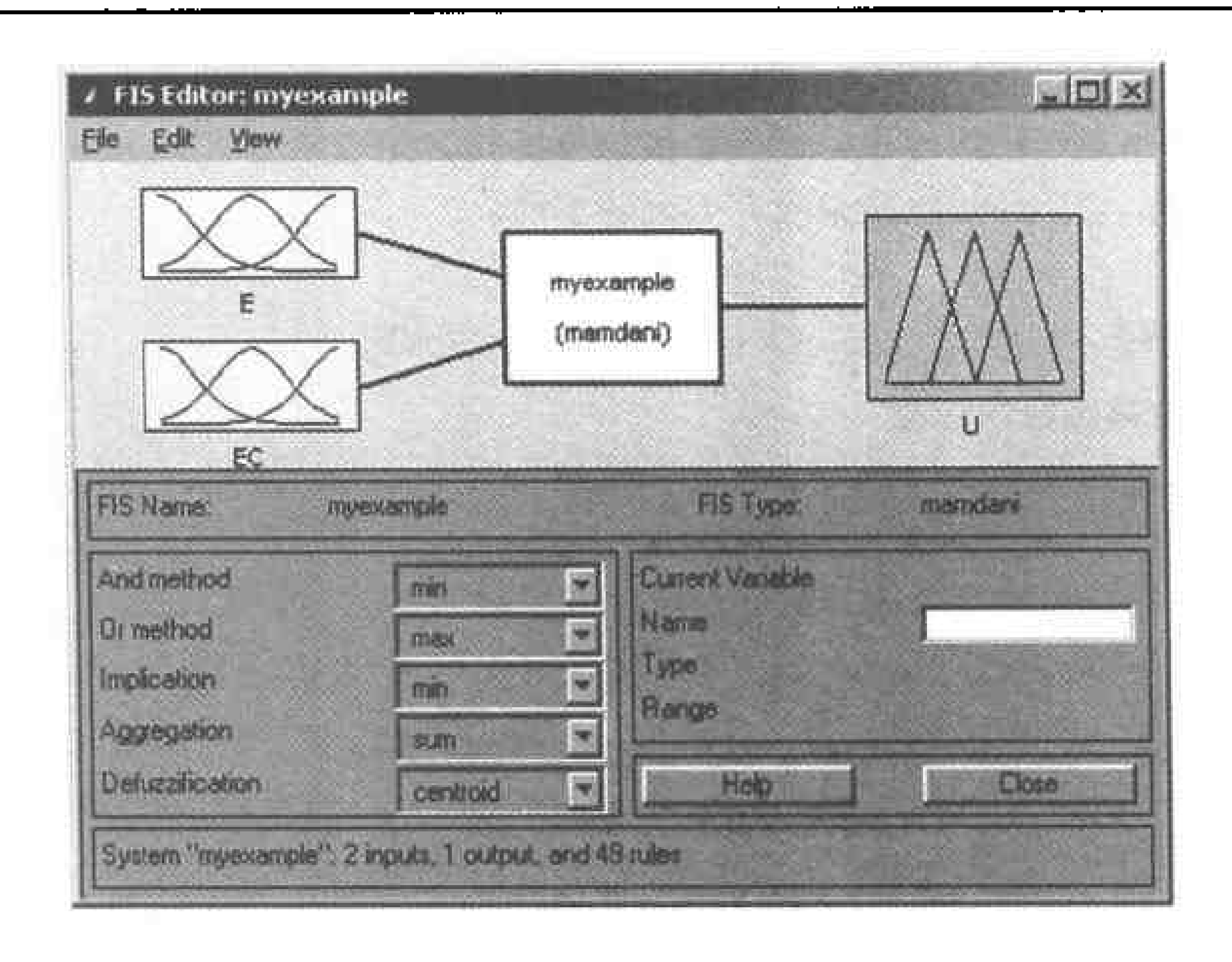


图 4－7 已建立的模糊推理系统

单击输入变量 E 的图标，然后在隶属函数编辑器的 Edit 下拉菜单中选择“Add MFs...”，弹出如图 4－8 所示的对话框。

按照前面的描述，选择变量 E 的隶属函数为高斯型隶属函数(共有三角形、高斯形、梯形等 11 种隶属函数可供选择)，由于变量 E 的语言值有 7 个(NB，NM，NS，ZO，PS，PM，PB)，因此选择隶属函数的个数为 7。

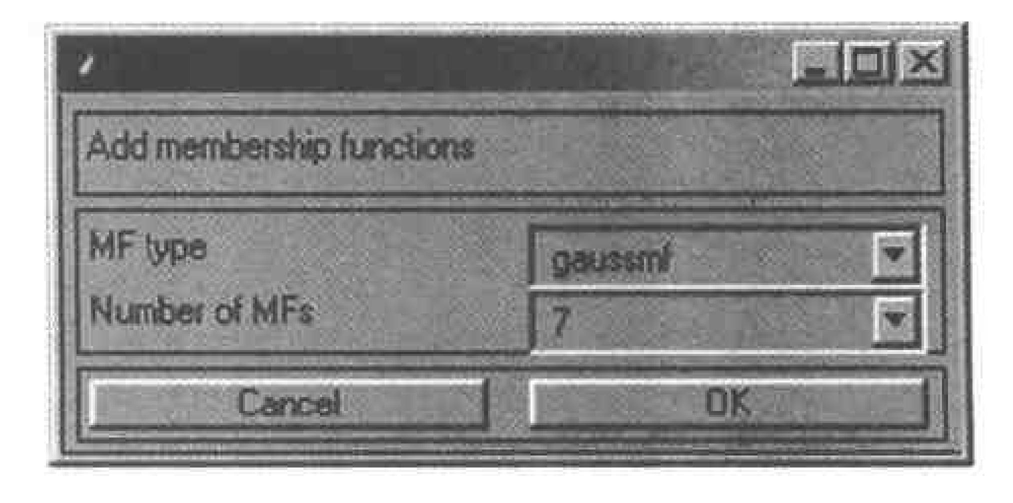


图 4－8 设置变量隶属函数的类型与数目的对话框

单击“OK”按钮，隶属函数编辑器的 GUI 界面被刷新。可以看到变量 E 被添加了 7 条高斯隶属函数，各隶属函数的名称分别为“MF1”到“MF7”。

在编辑器的左下的“Current Variable”框中，显示了当前变量的名称和类型，并可设置当前变量的变化范围和其在隶属函数上的显示范围。按照前面的描述，在“Current Variable”框的“Range”文本框和“Display Range”中，将变量 E 及其显示的范围均设置为[−6,6]。

在编辑器的隶属函数图形区单击选中一条隶属函数曲线，该曲线变为红色，表示成为当前隶属函数。在编辑器 GUI 界面右下区的“Current Membership Function”框中，可以修改当前隶属函数的名称、类型和参数。

在“Current Membership Function”框的“Name”文本框中，分别按变量 E 的语言值将各隶属函数命名为 NB,NM,NS,ZO,PS,PM,PB。

在“Current Membership Function”框的“Type”选择栏中，可以修改当前隶属函数的类型(不会影响到其它隶属函数的类型)。

在“Current Membership Function”框的“Params”文本框中，可以修改当前隶属函数的参数。更直观的方法是，用鼠标按住当前隶属函数并拖动鼠标，可以改变当前隶属函数的位置；用鼠标按住当前隶属函数上的小方框并移动鼠标，可以改变当前隶属函数的形状。“Params”文本框中的参数将实时地变化，以反映这些操作所赋予当前隶属函数的改变。

完成了如上的操作后，隶属函数编辑器上将完整地显示出输入变量 E 的隶属函数，如图 4-9所示。

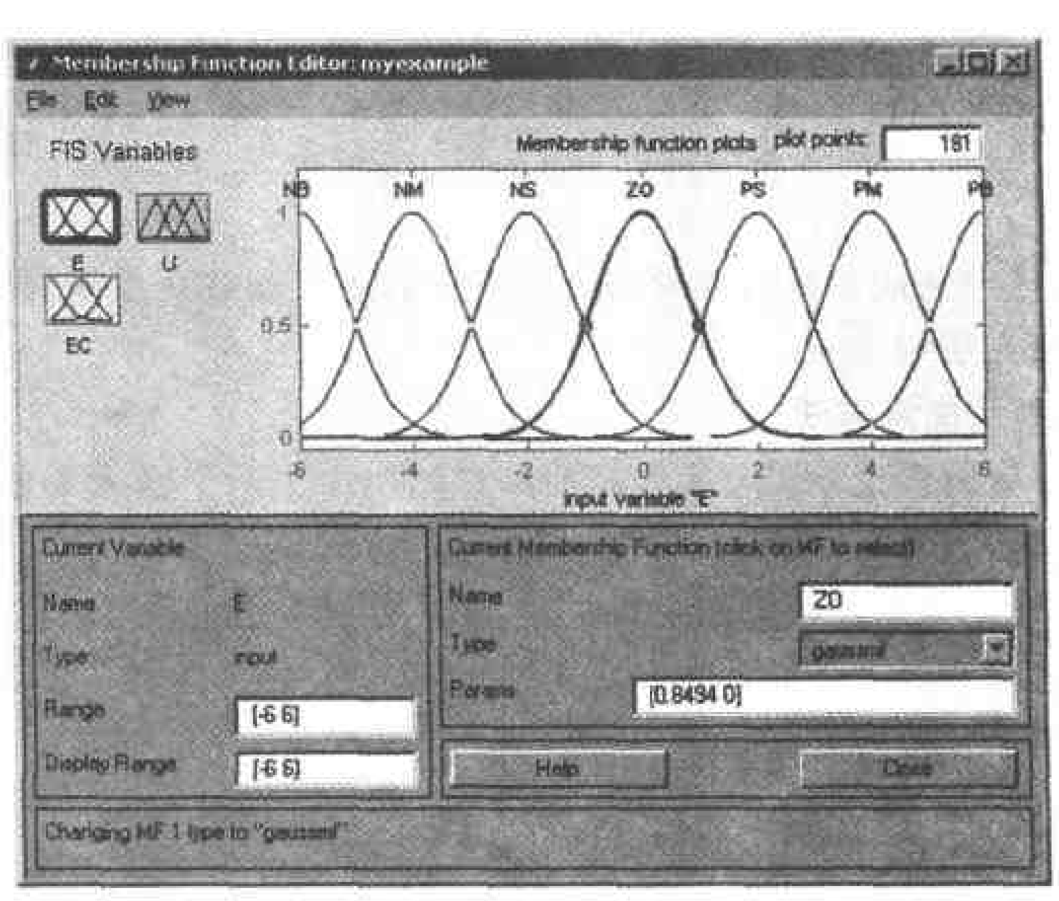
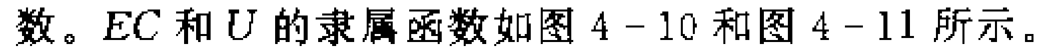


图 4-9　隶属函数图形及其修改

按照同样的方法与步骤，按照自己的经验，添加并设置输入 EC 和输出变量 U 的隶属函


数。EC 和 U 的隶属函数如图 4－10 和图 4－11 所示。

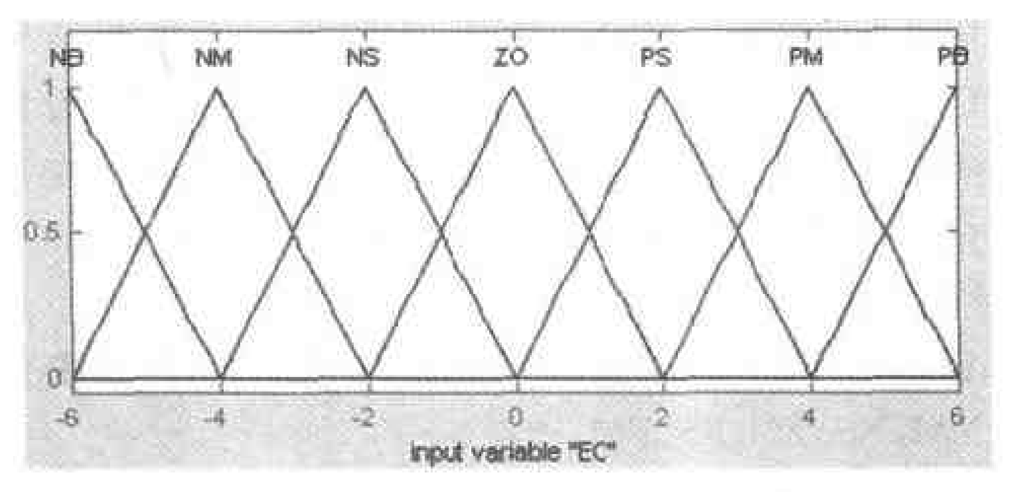


图 4－10　变量 EC 的隶属函数

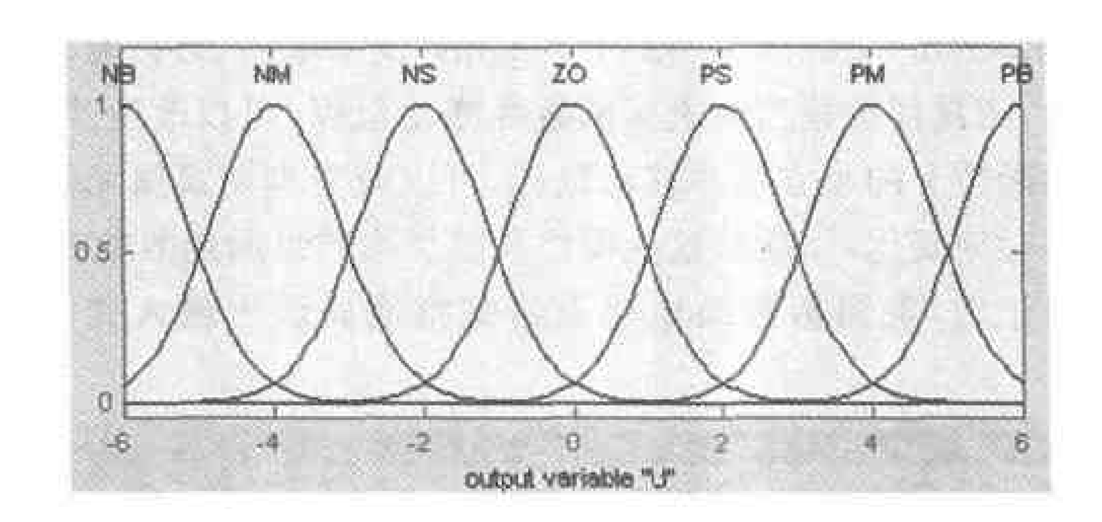


图 4－11　变量 U 的隶属函数

至此，实质上就已经完成了模糊推理系统中变量的定义和各变量隶属函数的设置。下面进行模糊规则和决策方法的建立。

2. 模糊控制规则及决策方法

控制规则是对专家的理论知识与实践经验的总结。由于两个输入变量各有 7 个语言值，因此共有 49 条规则，如表 4－1 所示。

为方便起见，一般按模糊控制规则表从左到右，从上到下的顺序输入模糊规则。在模糊控制规则表中看不到推理前提中的关系为“与”还是“或”，但在输入时各前提关系是可以选择的。如表 4－1 中，将规则 1 解释为

if E is NB and EC is NB then U is PB

下面在模糊规则编辑器窗口中输入这 49 条控制规则。双击 FIS 编辑器上部中央表示模糊推理系统的“myexample”白色方块（或从 FIS 编辑器的 View 下拉菜单中选择“Edit rules...”），将打开模糊规则编辑器。

表 4－1　模糊控制规则表

E \ U \ EC	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PB	PB	PM	ZO	ZO
NM	PB	PB	PB	PB	PM	ZO	ZO
NS	PM	PM	PM	PM	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NM	NM	NM	NM
PM	ZO	ZO	NM	NB	NB	NB	NB
PB	ZO	ZO	NM	NB	NB	NB	NB

选择模糊规则编辑器左边输入变量 E 的语言值为 NB，EC 的语言值为 NB，并在编辑器左下区的“Connection”框中选择输入变量 E 和 EC 的关系为“and”（默认），然后同时选择编辑器右边输出变量 U 的语言值为 PB，然后单击编辑器下方的“Add rule”按钮，则在编辑器的规则显示区给出了第一条模糊规则。如下所示。

if (E is NB) and (EC is NB) then (U is PB)(1)

在显示的规则末有一个(1)，它表示该规则的权值。在编辑器左下的“Weight”文本框中，可以修改当前规则的权。

按照同样的方法与步骤，输入第 2 条到第 49 条规则。模糊规则编辑器的显示如图4－12所示。

如果要改变某一条规则，只需在规则显示区中选择该规则使其变为蓝色，然后在编辑器中重新确定该规则的其它选项或参数，并按“Change rule”按钮即可。

要改变模糊决策和反模糊化方法，只须在 FIS 编辑器左下区域中改变“Aggregation”和“Defuzzification”的选项即可。至此，就完成了模糊规则和模糊决策的确定。

3. 观察模糊规则和模糊推理的输出曲面

在任一编辑器的 View 下拉菜单中选择“View rules...”，可以观察模糊规则的图形表示，如图 4－13 所示。

在任一编辑器的 View 下拉菜单中选择“View rules...”，可以观察模糊推理的输出曲面。用鼠标按住输出曲面并移动鼠标，可以从不同的角度观察输出曲面，如图 4－14 所示。

在任一编辑器的 File 下拉菜单中选择“Save to disk”，将所做的工作保存到磁盘中。

至此，就完成了整个模糊推理系统的建立。

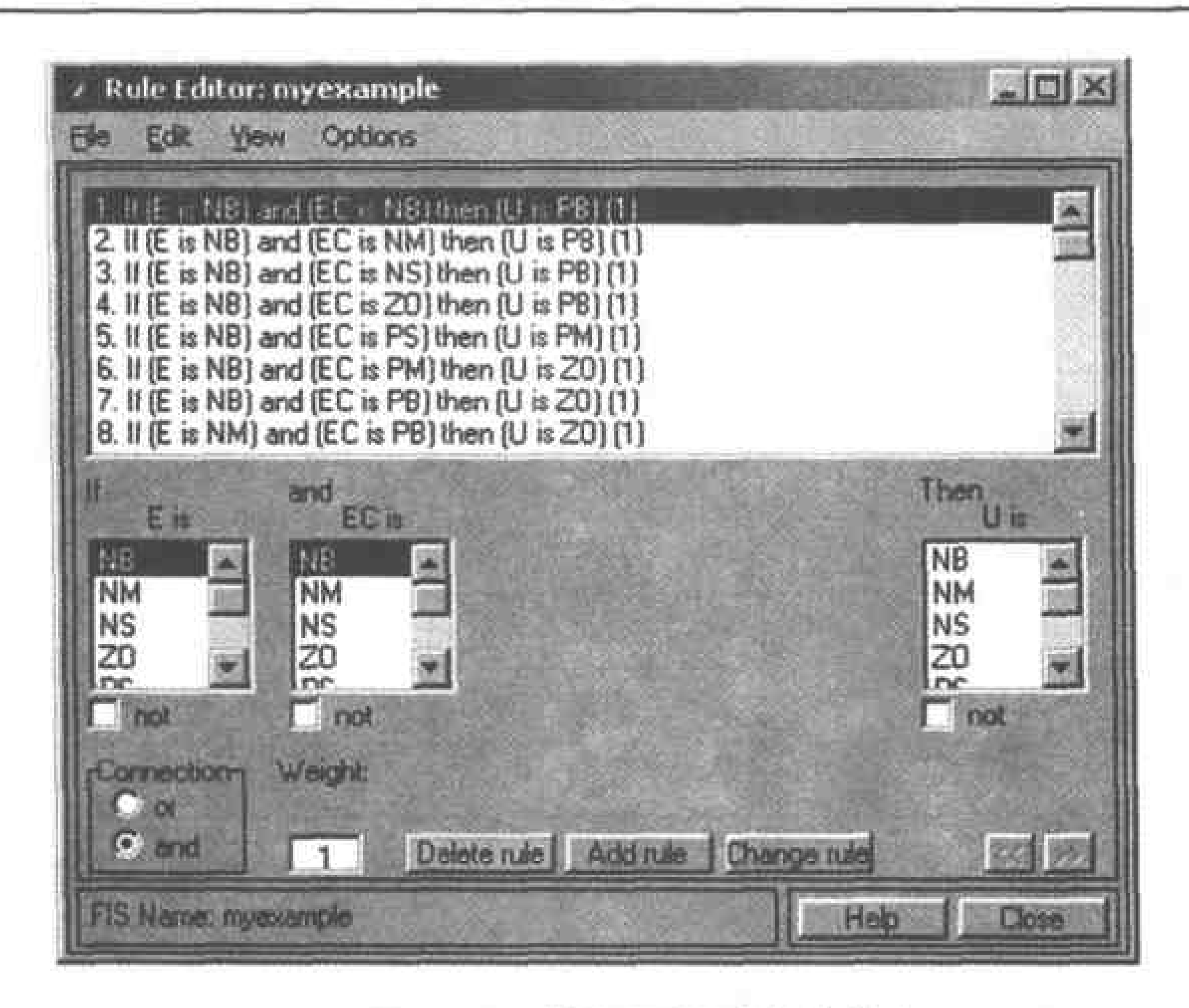


图 4－12　模糊规则的确定与修改

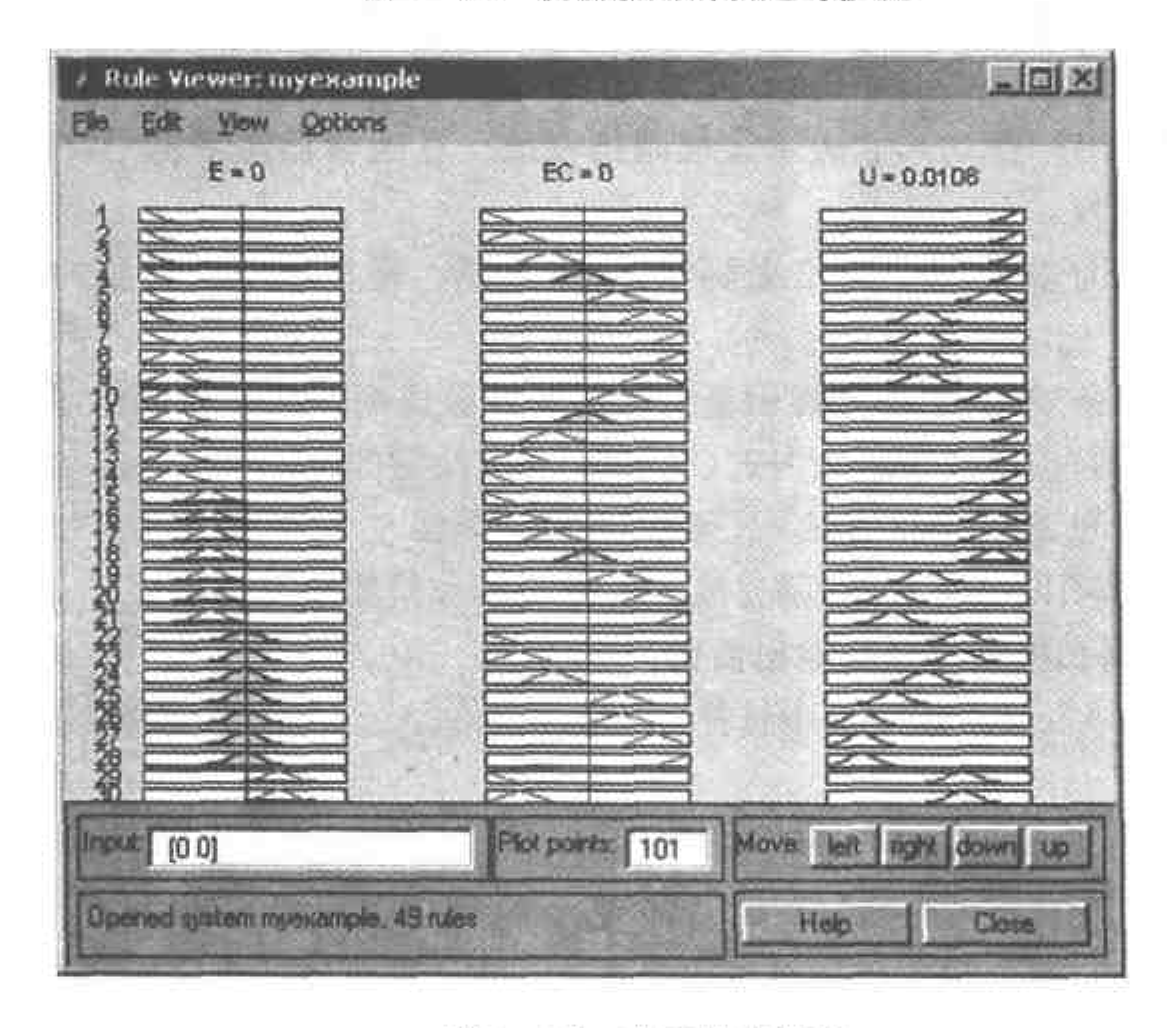


图 4－13　观察模糊规则

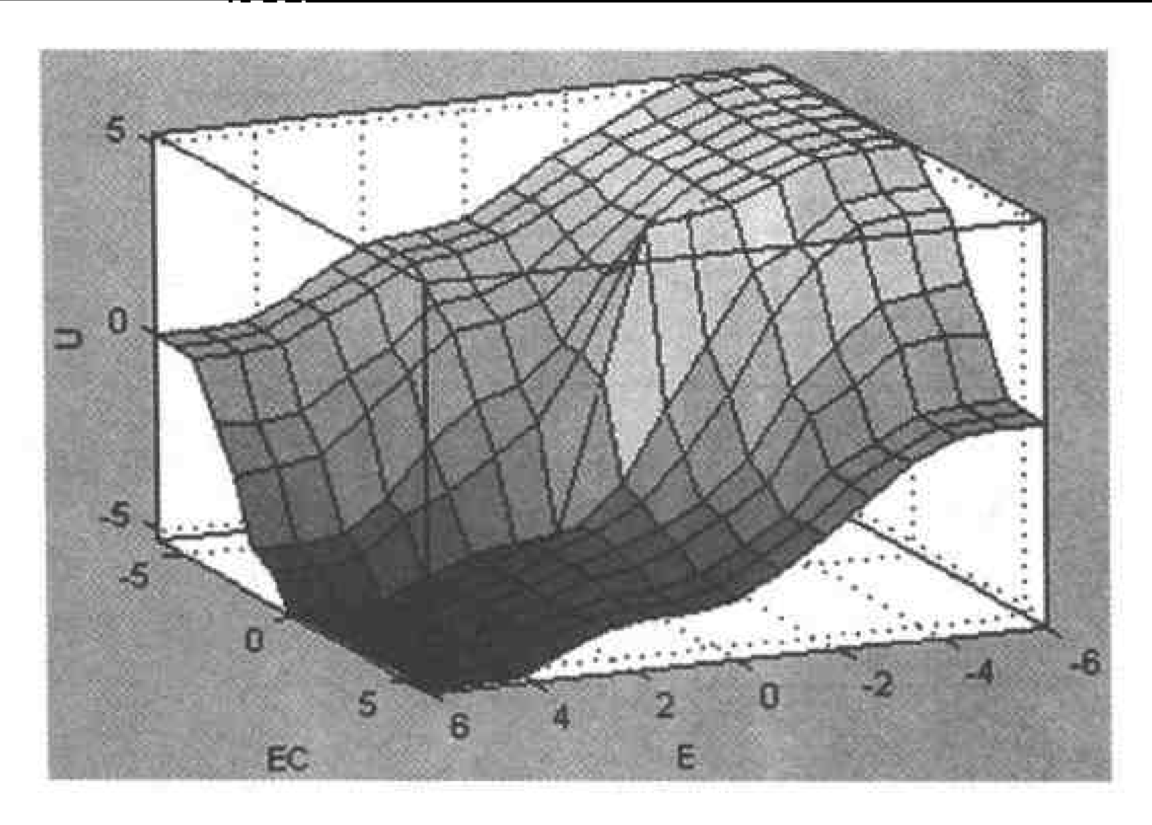


图 4-14 观察输出曲面

4.2.2 Simulink 仿真及参数的调整

如 4.1 中所述，对前面建立的模糊推理系统，首先单击模糊逻辑工具箱 5 个 GUI 界面中任一个的 File 菜单，选择其中的“Save to workspace”，将其保存到 MATLAB 的工作区中。

其次，在 MATLAB 工作区中键入命令“simulink”，然后在打开的 Simulink 库中建立自己的 Simulink 仿真模型。对采用的模糊控制方法，在 Simulink 库的“Fuzzy Logic Toolbox”项中，选择“Fuzzy Logic Controller”作为我们的模糊控制器模块。在仿真模型中双击“Fuzzy Logic Controller”的图标，在弹出的对话框中的“FIS Matrix”文本框中输入前面保存到 MATLAB 的工作区中的模糊推理系统的名称，将工作区中的 FIS 结构与该模糊控制器连接起来。

然后，选择向量信号组合工具“Mux”，将其输出与模糊逻辑控制器相连，其输入则与 FIS 结构的输入变量相连。双击其图标，在弹出的对话框中确定输入的个数为 2。

至此，完成了模糊仿真模型的建立，如图 4-15 所示。事实上，这是一个由模糊控制器与一阶滞后被控对象组成的采样控制系统。

在 Simulation 下拉菜单中选择“Start”，开始仿真。双击仿真模型中的显示器“Scope”，将实时显示出系统的输出。如果仿真的控制效果不满意，一般首先调整比例变换因子 K_e，K_{ec} 及 K_u，再调整模糊控制规则和隶属度函数。

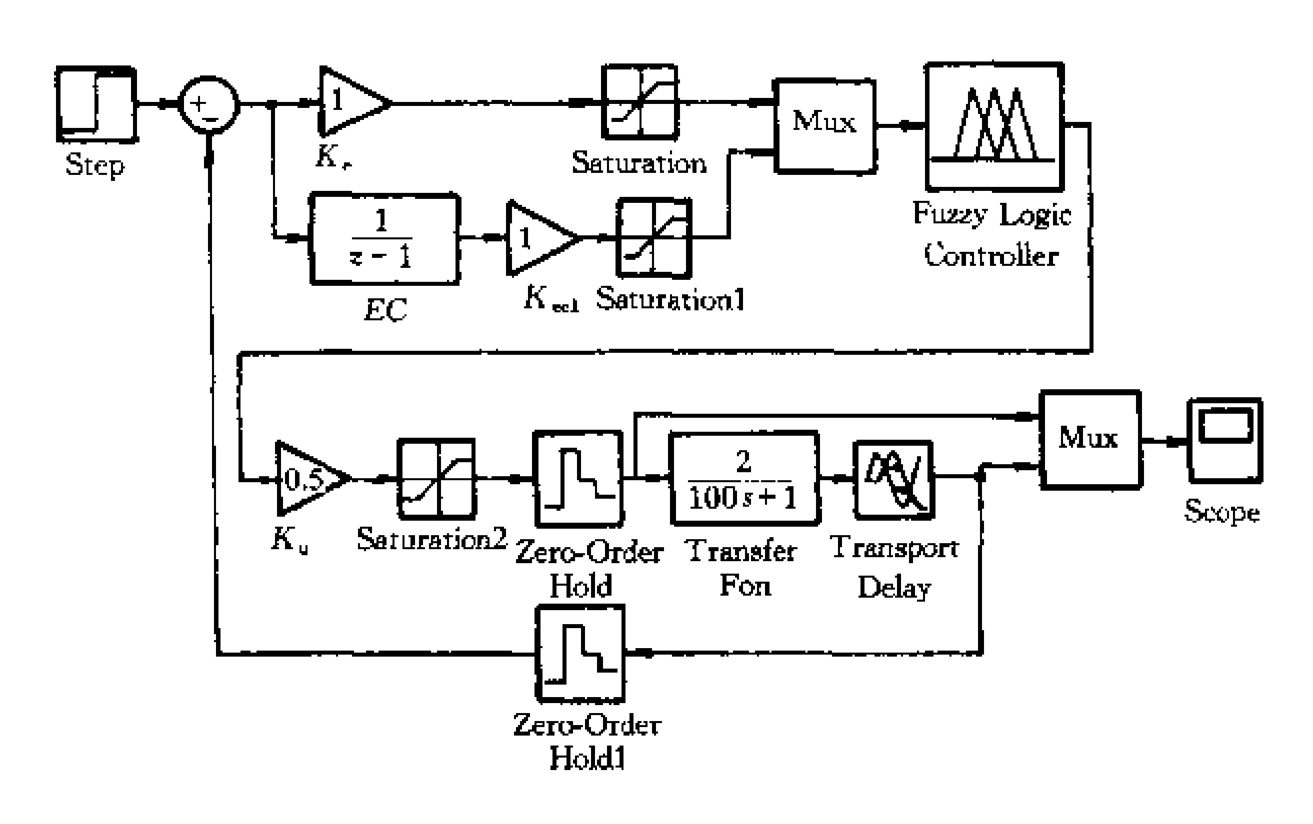


图 4-15　模糊 PD 控制仿真系统

4.2.3　仿真结果

图 4-15 所示系统中的被控对象为

$$G(s)=\frac{2}{100s+1}$$

采样周期为 1s。通过仿真可看出一般的模糊控制器相当于非线性的 PD 控制，无积分作用，有静差；通过调整系统参数可见，在比例变换因子中，K_u 对系统稳定性影响最大，过大时系统失稳。图 4-16、图 4-17 为系统阶跃输入下，K_u 适当时被控对象稳定的输出响应和 K_u 过大时被控对象失稳的输出响应。图中深灰色曲线为系统响应，浅灰色曲线为模糊控制器输出的控制量。

计算机仿真试验对实际控制有重要指导意义。由上述模糊控制器的设计和仿真过程可看出 MATLAB 的模糊逻辑工具箱及 Simulink 有如下特点：

①可灵活地设计模糊控制器。在模糊逻辑工具箱中能方便地修改输入输出的论域、模糊子集、隶属度函数、模糊控制规则、模糊决策方法及反模糊方案，然后通过仿真找到较佳的控制方案。

②可直接计算出模糊控制表，供单片机进行实际控制时使用，而人工计算此表非常繁琐。

③在 Simulink 环境下，可组成更复杂的控制系统，如对象为时变、非线性，控制器为模糊与 PID 的混合形式。

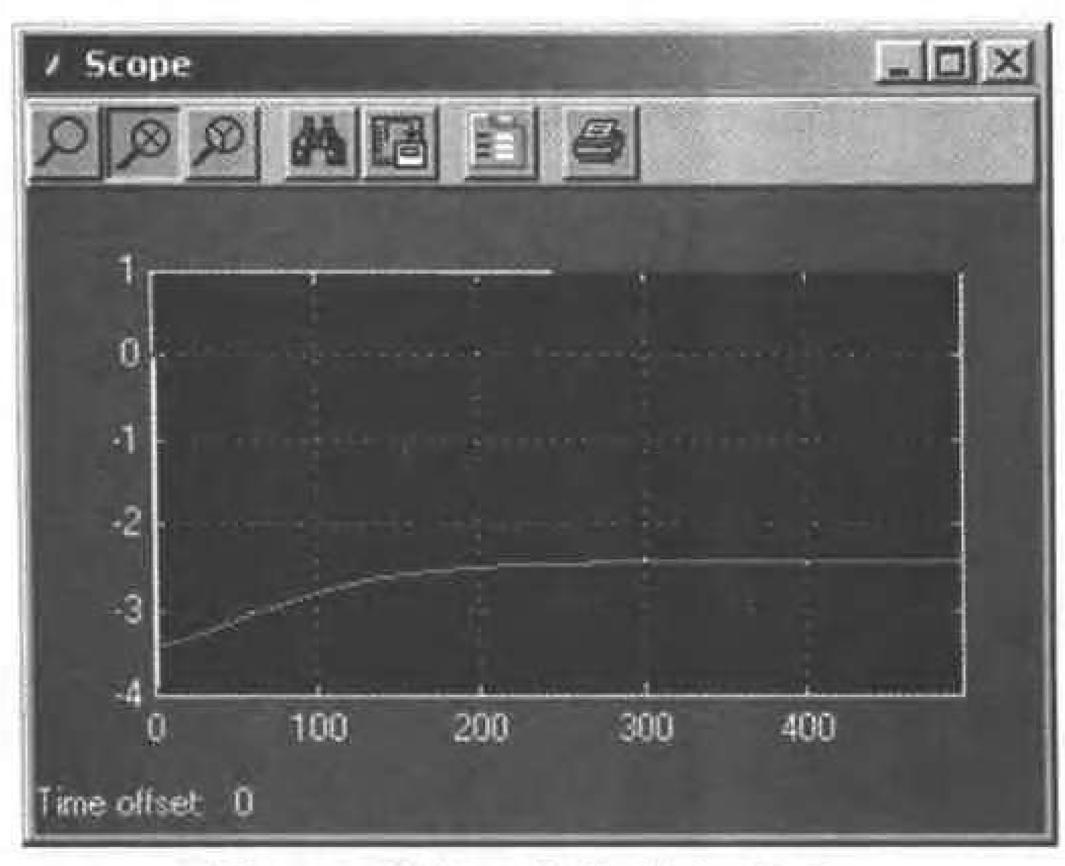


图 4-16 模糊 PD 控制系统仿真结果 1

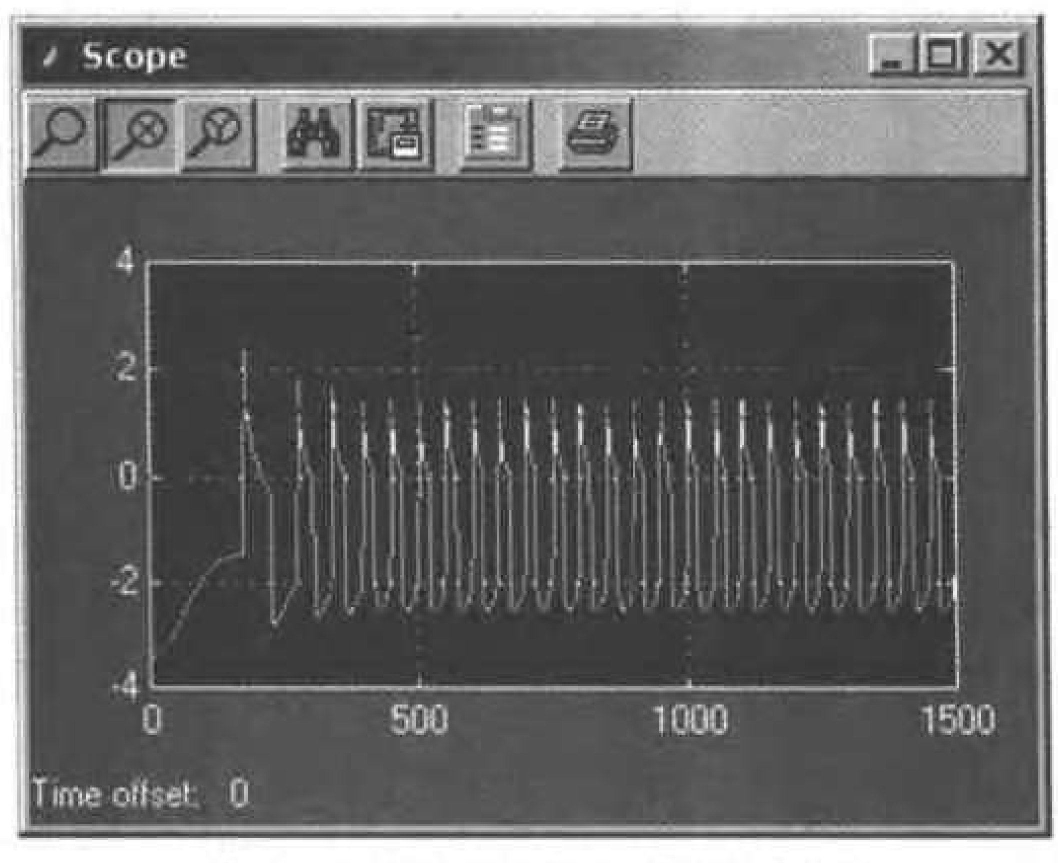


图 4-17 模糊 PD 控制系统仿真结果 2

4.3 火电机组主汽温模糊控制系统

1. 串级 PID 调节火电机组主汽温控制

对火电机组主汽温控制系统进行模糊控制设计。目前国产火电单元机组主汽温调节普遍采用串级 PID 调节方式，扰动主要考虑减温水的自发扰动 $f(t)$ 和燃烧率的改变 $g(t)$ 两个因素。利用 MATLAB 中的 Simulink 平台组建的 PID 串级调节系统方框图如图 4－18 所示。在 Simulink 界面中可以用鼠标双击模型，在对话框中填入各个模型的参数。

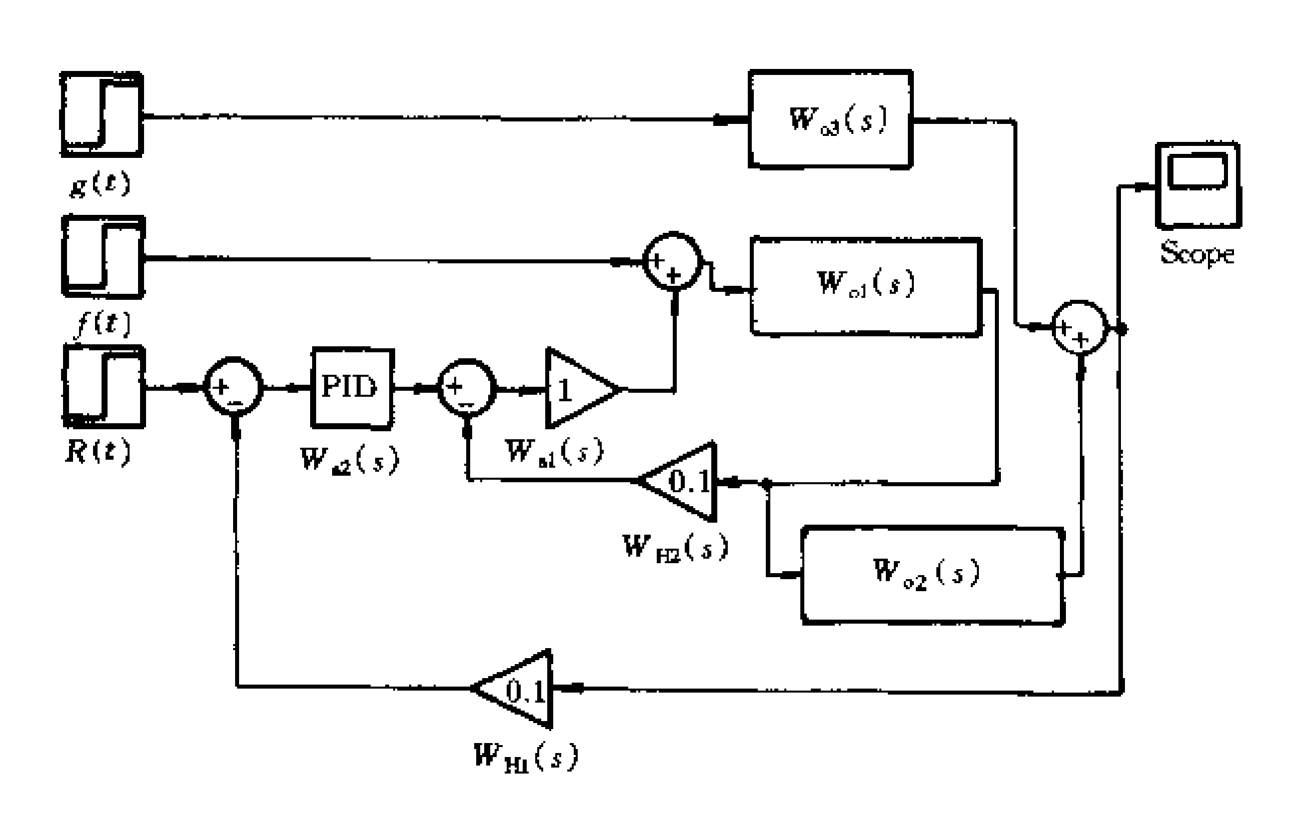


图 4－18 火电机组主汽温控制串级 PID 调节仿真系统

图 4－18 中，$W_{a1}(s)$，$W_{a2}(s)$ 分别为串级调节系统中的副调节器和主调节器；$W_{o1}(s)$，$W_{o2}(s)$ 分别为调节对象及其导前区的传递函数；$W_{o3}(s)$ 是燃烧率扰动对主汽温的传递函数；$W_{H1}(s)$，$W_{H2}(s)$ 分别为导前汽温和过热汽温的测量单元。被控对象模型为：

$$W_{o1}(s)=\frac{8}{(1+15s)^2}=\frac{0.035556}{(s+0.6667)^2}$$

$$W_{o2}(s)=\frac{1.125}{(1+25s)^3}=\frac{0.000072}{(s+0.04)^3}$$

$$W_{o3}(s)=\frac{1}{1+20s}=\frac{0.05}{s+0.05}$$

PID 控制器 $W_{a2}(s)$ 的比例、积分和微分系数分别设置为 1，2 和 1。$W_{a2}(s)$ 设置为 1。$R(t)$，$g(t)$ 和 $f(t)$ 均设置为幅值为 1 的阶跃输入。$W_{H1}(s)$，$W_{H2}(s)$ 的比例系数均设置为 1。

设置仿真时间为 1 500 s。则得到系统的输出如图 4－19 所示。

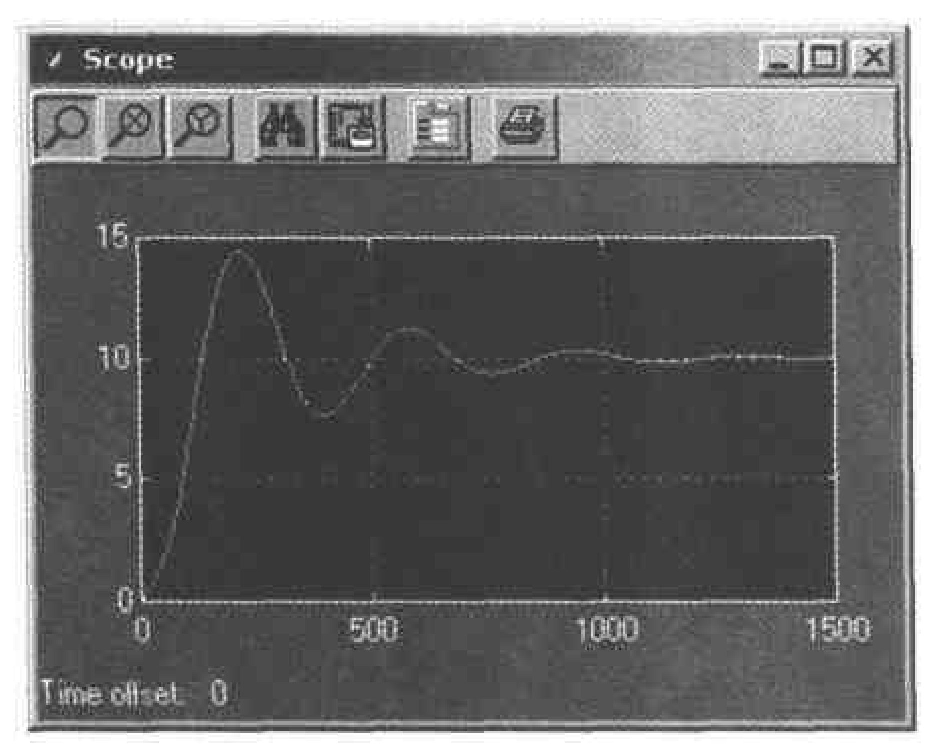


图 4－19　主汽温控制串级 PID 调节系统仿真结果

2. 模糊控制器设计

利用经典模糊控制器取代原 PID 调节器，副调节器仍然采用一个放大环节。在 Simulink 平台下构建该模糊控制器的结构图，如图 4－20 所示，利用 Fuzzy Logic 工具箱进行经典模糊控制器的设计，并将计算完毕得到的矩阵变量赋给图 4－20 中的 Fuzzy 模块。

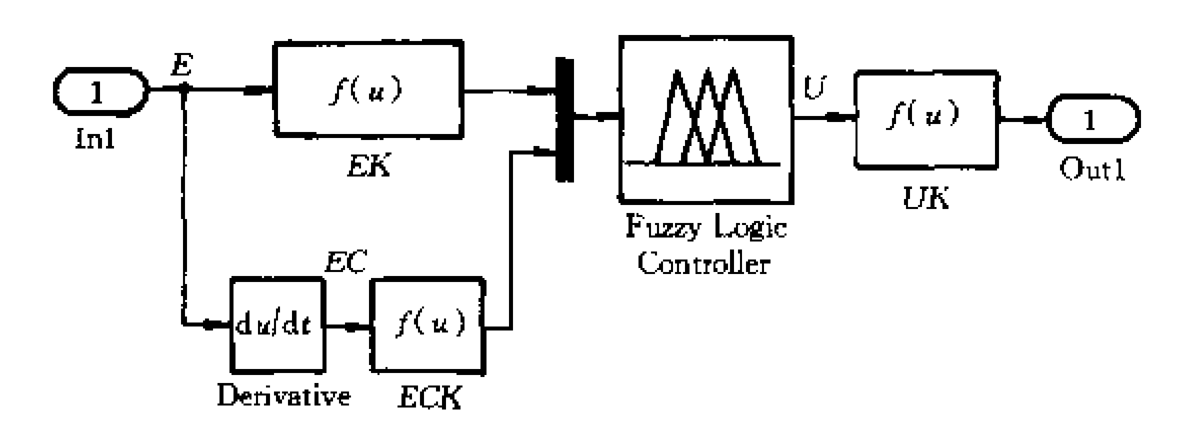


图 4－20　主汽温控制模糊控制器

图 4－20 中，E，EC 和 U 分别为误差 e、误差变化 de/dt 和控制量 u 的模糊语言变量，EK，ECK 和 UK 为量化因子。模糊控制器设计参数如下：

(1) 语言变量的模糊集

E＝{NB，NM，NS，ZO，PS，PM，PB}

EC＝{NB，NM，NS，ZO，PS，PM，PB}

U＝{NB，NM，NS，ZO，PS，PM，PB}

(2) 模糊变量论域、隶属函数类型及反模糊方法

E,EC 和 U 的论域皆为[-20,20],隶属函数为三角类型。解模糊化方法采用中位数法。

(3) 模糊规则的描述

该模糊控制器的模糊控制规则具有以下形式:

if{$E=A_i$ and $EC=B_i$} then $U=C_i$, $i=1,2,\dots,n$

其中 A_i,B_i 以及 C_i 都是各自论域上的模糊语言值(如 NB、NS 等),其模糊控制规则表如表 4-2 所示。

表 4-2　主汽温控制模糊规则表

E \ U \ EC	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PB	PB	PM	ZO	ZO
NM	PB	PB	PB	PB	PM	ZO	ZO
NS	PM	PM	PM	PM	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NM	NM	NM	NM
PM	ZO	ZO	NM	NB	NB	NB	NB
PB	ZO	ZO	NM	NB	NB	NB	NB

可按前述 Verbose 规则编写格式将模糊控制规则表中的值一一写入模糊控制规则编辑器中。

模糊推理的输出曲面如图 4-21 所示。

3. 模糊串级控制系统设计

将量化因子取 $EK=5$,$ECK=320$,$UK=1/3.5$(因子可在仿真过程中调整),在 Simulink 平台下建立整个模糊串级控制系统仿真方框图,如图 4-22 所示。

由于常规模糊控制器具有 Fuzzy 比例-微分作用,缺少积分环节,因此必然不可避免地产生稳态误差,而电站主汽温控制系统对稳态精度要求很高,因此在图中引入 PI 控制策略,构成 Fuzzy-PI复合控制器。

4. 仿真结果比较

Simulink 中的 Sinks 工具库提供了诸如 Scope,XY Graph 等仿真曲线绘制工具模块,如同示波器一样,可以非常快速直观地得到动态响应仿真结果。同时也可以选择 Workspace 模块,将输出结果以变量形式存入内存,再在 MATLAB 编程环境中编程绘图或进行数据文件 I/O操作。

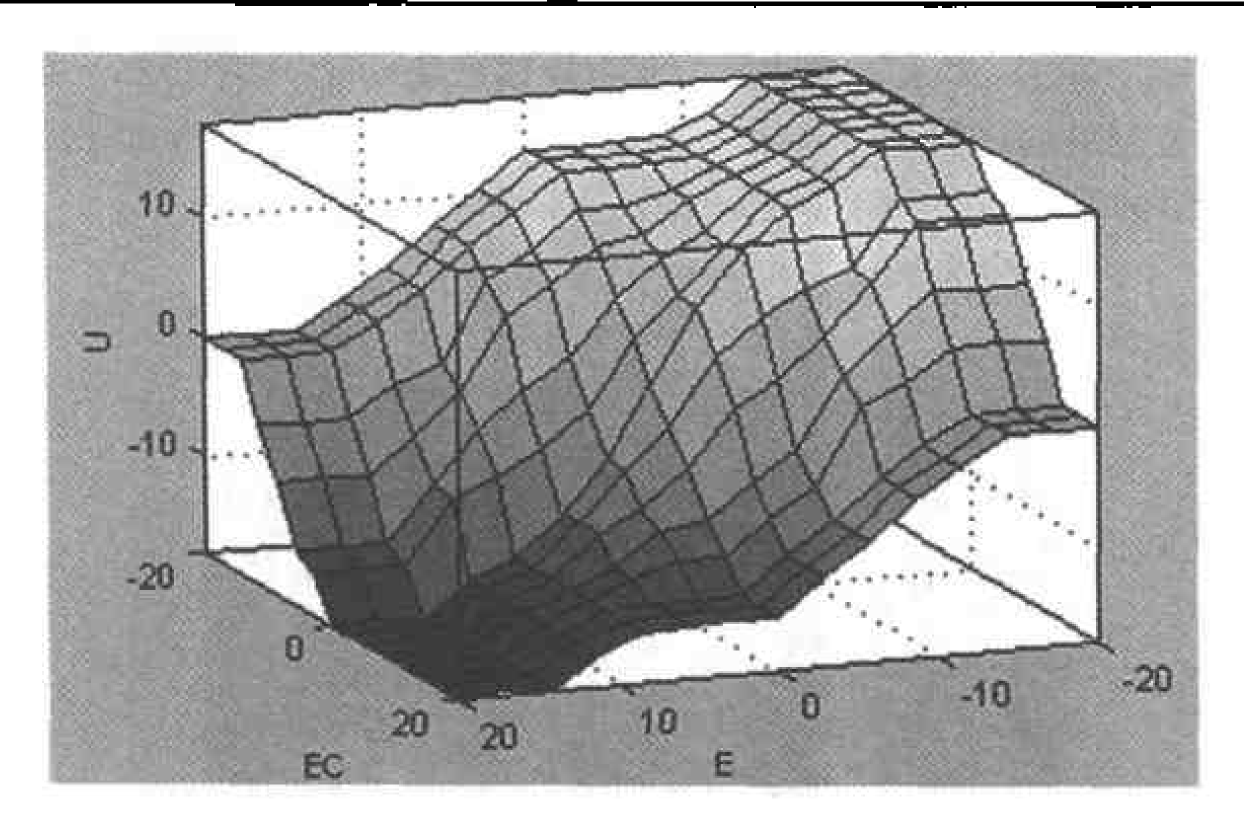


图 4-21　主汽温控制模糊推理输出

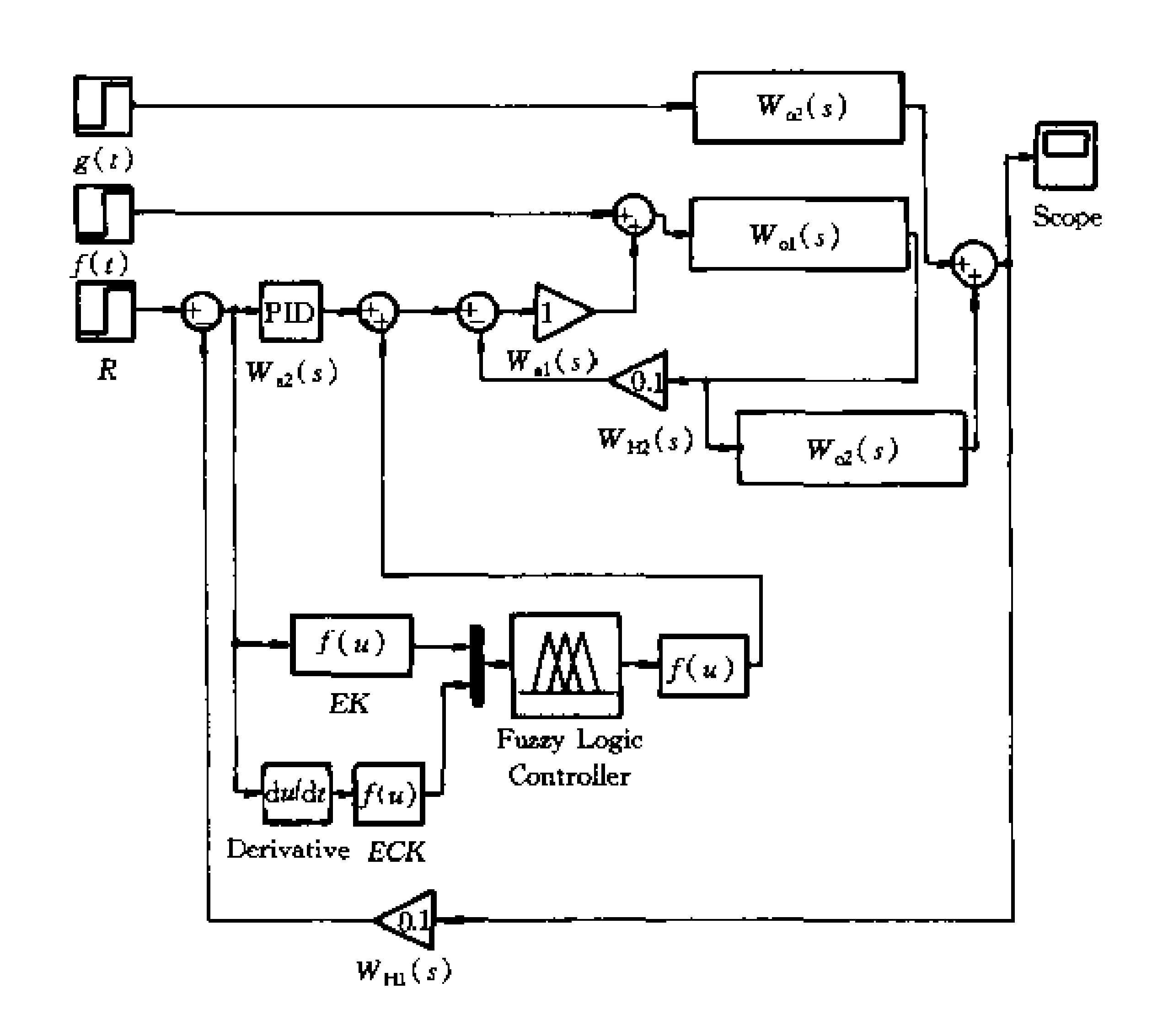


图 4-22　主汽温模糊控制仿真系统

假定取 $R=1\text{mA}$，在 $t=500\text{s}$ 时加入减温水自发阶跃扰动 $f(t)=10t/\text{h}$。在串级 PID 控制系统仿真中，PID 控制器的比例、积分和微分系数分别设为 1，1 和 0，将仿真时间设为 1 500 s，利用 Simulink 所绘的系统动态响应曲线如图 4－23 所示。

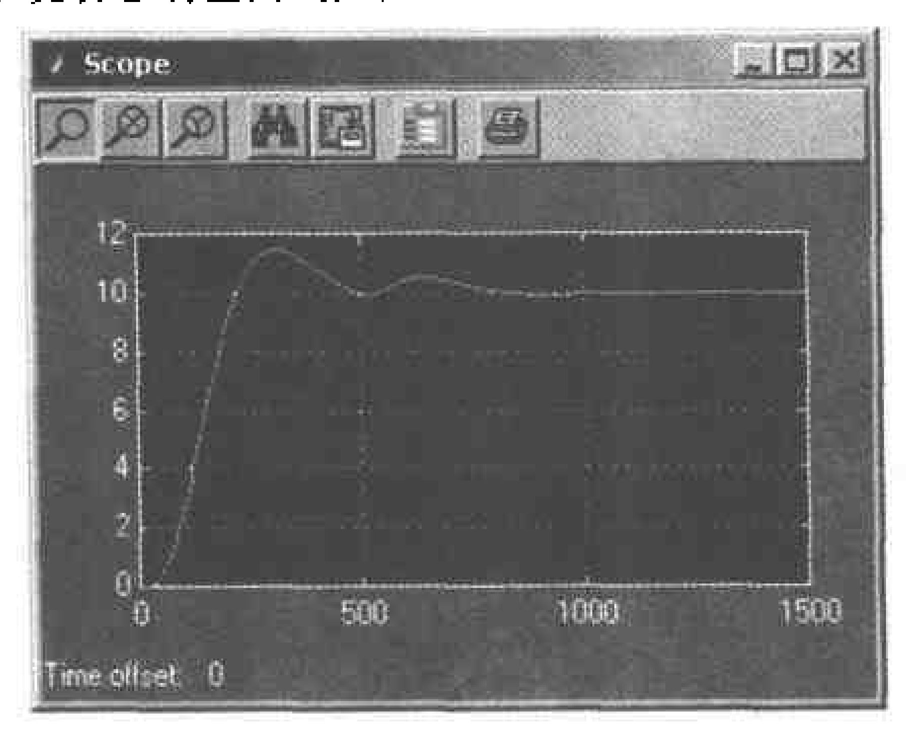


图 4－23 动态响应曲线 1

图 4－23 中的输出曲线为 Fuzzy－PI 串级调节的动态响应曲线。由仿真结果可以看出，采用主调节器为 Fuzzy-PI 调节器的串级调节系统，具有较强的抗干扰能力，调节品质良好。

利用同样的方法可以在 Simulink 下绘制出燃烧率扰动时的动态响应曲线，假定 $t=500\text{s}$ 时加入 $g(t)=10$（即 10％的燃烧率扰动），可得如图 4－24 所示的动态响应曲线。

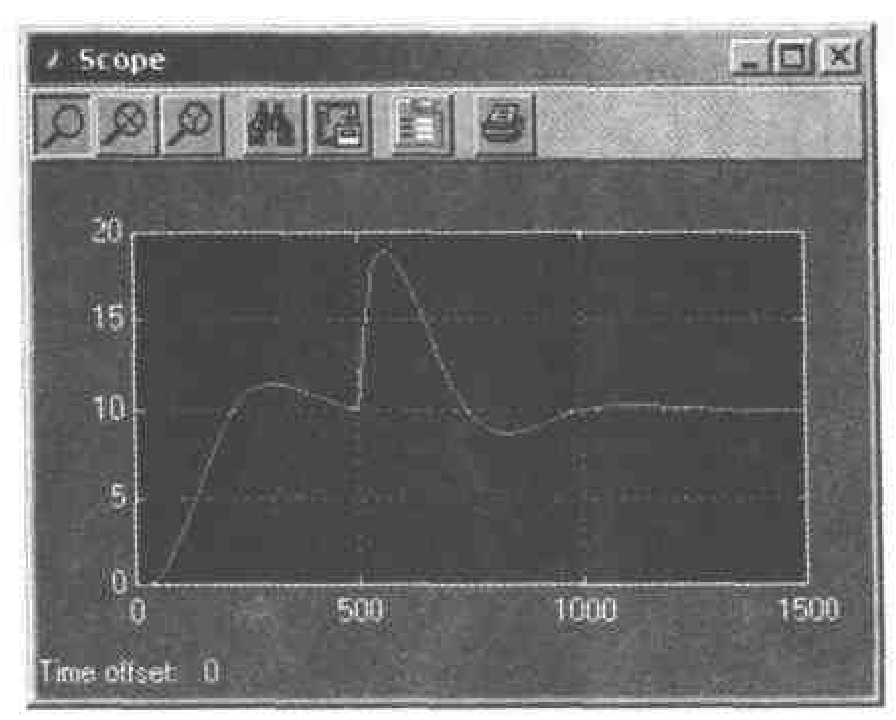


图 4－24 动态响应曲线 2

假定 $t=500\text{s}$ 时同时加入10%的燃烧率扰动 $g(t)=10$ 和减温水自发阶跃扰动 $f(t)=10t/\text{h}$，得到的系统响应如图4-25所示。

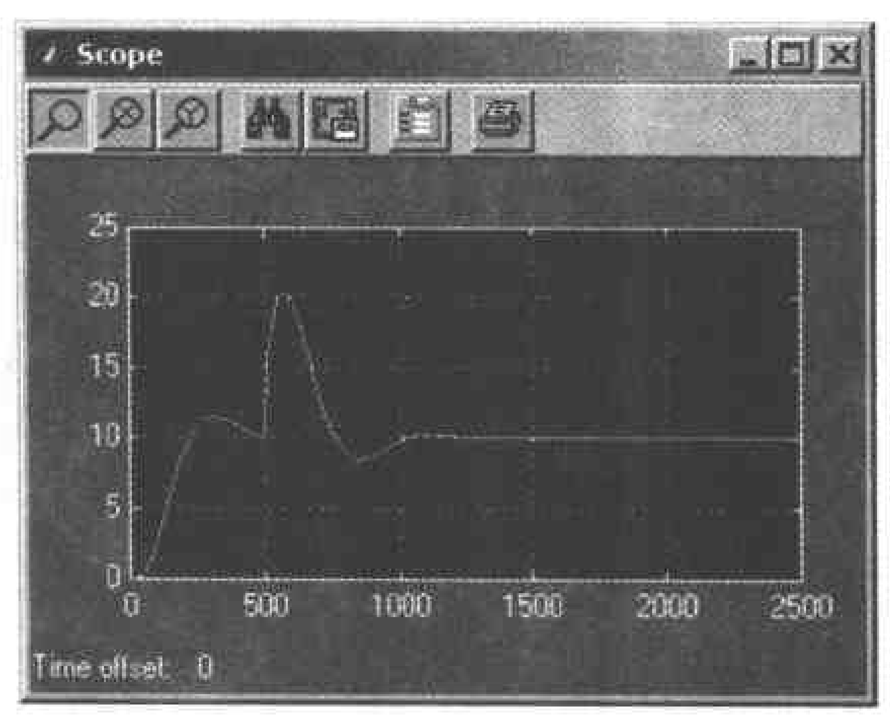


图4-25　动态响应曲线3

5. 结论

从以上仿真实例可以看出，利用MATLAB来构建模糊控制器和模糊控制系统简单快速，精度高，可视性强，从建模到最终的动态曲线分析，用户都不需要编写计算机程序，显示出了MATLAB功能的强大。需要指出的是，模糊控制的类型种类很多，诸如：多模模糊控制器、规则可调模糊控制器、量化因子自调整模糊控制器、三维模糊控制器、模糊PID控制器等。在此仿真实例中所用Fuzzy－PI控制只是模糊控制方法中的一种，而且从图4-23至图4-25动态响应曲线中可看出还存在不理想的地方，如反向超调等。如何利用MATLAB中功能繁多的模型库和工具箱，设计组合成结构更合理、性能更优的模糊控制器，需要进一步研究。

4.4　单隶属度函数模糊控制器

由于大多数被控对象具有非线性、时变等特性，在模糊控制器中加入自组织功能是非常重要的。但是在模糊控制器结构设计中，论域范围的确定、量化、语言变量分档、隶属度函数的定义及控制规则的确定等诸多环节都蕴含着人为的因素，一般的自组织模糊控制器只能对其中的一个或几个环节进行调整，很难使整个控制器处于最优状态。若能设计出一种模糊控制器，从输入量精确值到输出量精确值可用一数学表达式来表达，则系统运行时只需在线调整算式中的某些参数即可。在这种设想下，提出了单隶属度函数模糊控制的方法。

1. 原理和思路

对于一个典型的反馈系统，从操作工的角度来说，有一套基本的控制规律，即观察到偏差

较大时，增强控制作用；偏差较小时，减弱控制作用。对不同的被控对象，其“大”与“小”的含义是不同的，也就是说不同的被控对象对应着不同的描述其语言变量“大”“小”含义的曲线，见图 4－26。在系统运行过程中，可以通过对系统输入输出数据的计算寻找出一条最优的曲线来。将图 4－26中的曲线称为语言变量的“大小”隶属度，记偏差 e 的“大小”隶属度为 $M_e(e)$，偏差变化率 Δe 的“大小”隶属度为 $M_{ec}(\Delta e)$，控制量 u 的“大小”隶属度为 $M_u(u)$。

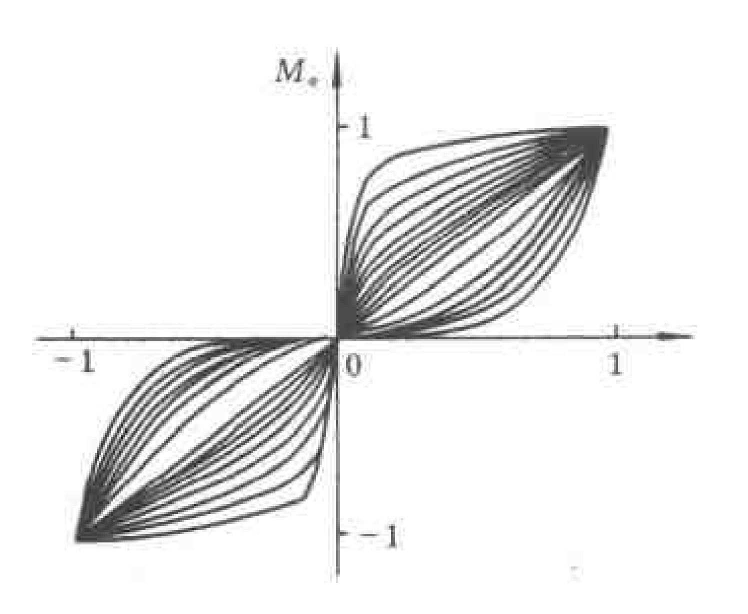


图 4－26　“大小”隶属度的定义

在系统运行过程中，既要对 $e,\Delta e,u$ 的“大小”隶属度进行调整，又要对控制算式进行调整。

为了能够对 $e,\Delta e,u$ 的“大小”隶属度进行调整，可将图 4－26 中曲线形状用 $y=x^p$ 的形式表达，多次试验表明，用这一表达式基本可以描述图 4－26 中的所有情况，且式中只有一个参数，便于寻优。为了能够对控制算式进行调整，将控制算式表达为：

$$M_u=f(M_e,\ M_{ec})$$

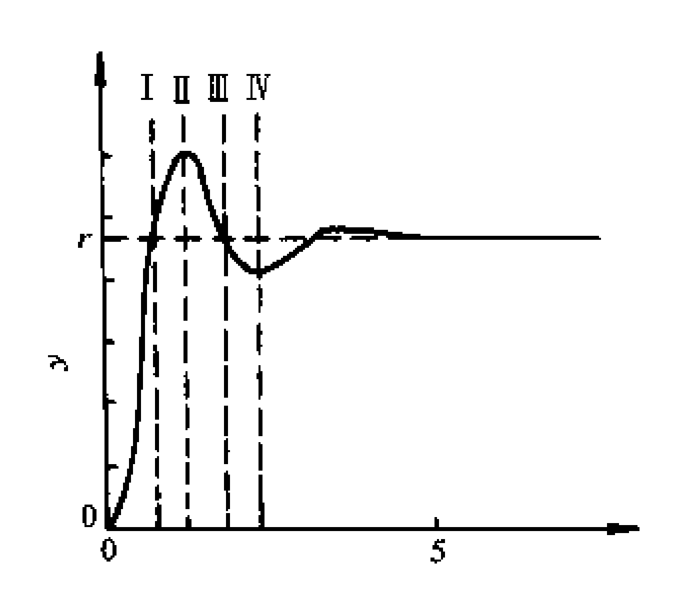


图 4－27　系统响应曲线

对图 4－27 所示的响应曲线Ⅰ，Ⅱ，Ⅲ，Ⅳ段分别讨论。定义 $e=r-y$，则：

① $e<0,\Delta e>0,M_u$ 应为正。

且 e 越负，M_u 应越大；Δe 越小，M_u 应越大。

即 $M_u=k_1|M_e|-k_2|M_{ec}|,k_1,k_2>0$。

② $e>0,\Delta e>0,M_u$ 应为负。

且 e 越大，M_u 应越负；Δe 越大，M_u 应越负。

即 $M_u=-k_3|M_e|-k_4|M_{ec}|,k_3,k_4>0$。

③ $e>0,\Delta e<0,M_u$ 应为负。

且 e 越大，M_u 应越负；Δe 越负，M_u 应越大。

即 $M_u=-k_5|M_e|+k_6|M_{ec}|,k_5,k_6>0$。

④ $e<0,\Delta e<0,M_u$ 应为正。

且 e 越负，M_u 应越大；Δe 越负，M_u 应越大。

即 $M_u=k_7|M_e|+k_8|M_{ec}|,k_7,k_8>0$。

综合上述 4 种情况，并考虑符号，可将控制算式写为：

$$M_u=k_eM_e+k_{ec}M_{ec}$$

为使后面的寻优过程简单，可将系数 k_e，k_{ec} 合并为一个参数 α，即

$$M_u = \alpha M_e + (1-\alpha) M_{ec}$$

α 代表着偏差与偏差变化率的权重。

寻优的目标函数 J 取为：

$$J = \sum_{k=1}^{N} k \mid e(k) \mid$$

由于被控对象的数学模型是未知的，很难推导出 J 与参数 p 和 α 之间的数学表达式，因此采用自寻优的方法。为简化自寻优过程，令 $M_u = u$，且 e 和 Δe 的"大小"隶属度表达式采用同一参数 p。通过计算机仿真研究得到 p 与 J 的函数关系，如图 4-28 所示。p 较小时，M_e 对小偏差很敏感，系统不稳定，极易振荡；p 较大时，M_e 对大偏差较敏感，对小偏差不够敏感，会引起过大超调，且回复时间长。因此存在一最优值 $p*$，使得 J 最小。为此采用步进搜索法寻优参数 p，但有些情况下得到最优值 $p*$ 后，仍得不到满意的输出曲线，还需寻优参数 α。所以说，这一算法的自寻优环节是一个二维的寻优过程，采用各参数依次寻优的方法，即先寻优参数 p，再寻优参数 α。

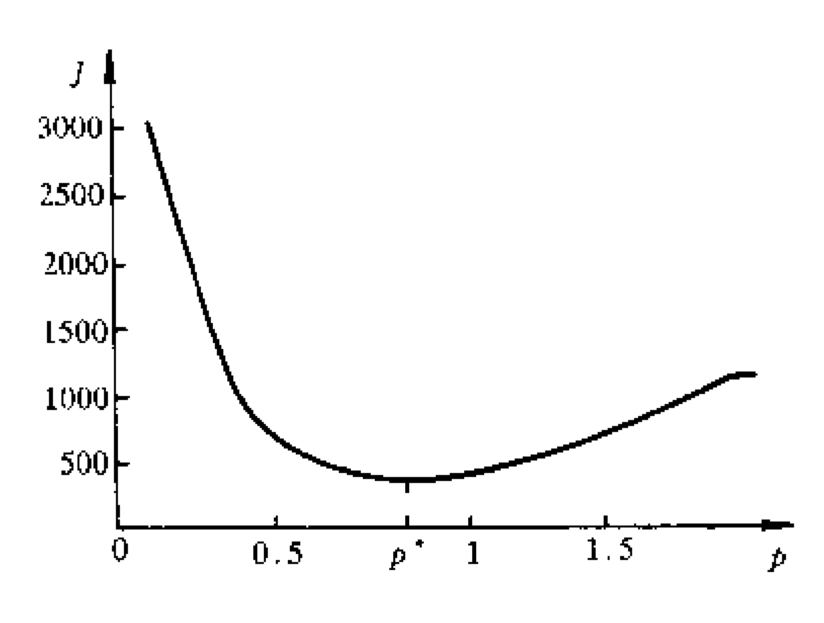


图 4-28　p 与 J 的函数关系

将上述思想用结构图的形式表达出来如图 4-29 所示。由于这种方法没有涉及语言变量的分档，只定义了一个衡量语言变量大小程度的"大小"隶属度函数，因此称之为单隶属度函数模糊控制方法。

2. 实现方法

以一个 SISO 系统为例：

① 将偏差、偏差变化率和控制量的论域转换到区间[−1，+1]上；

② 设定初始的 p_0 和 α_0 值，如果对被控对象一无所知，可设 $p_0=1$，$\alpha_0=0.5$；

③ 运行一个控制过程，记录下参数 p_i，α_i，并计算 J_i 值；

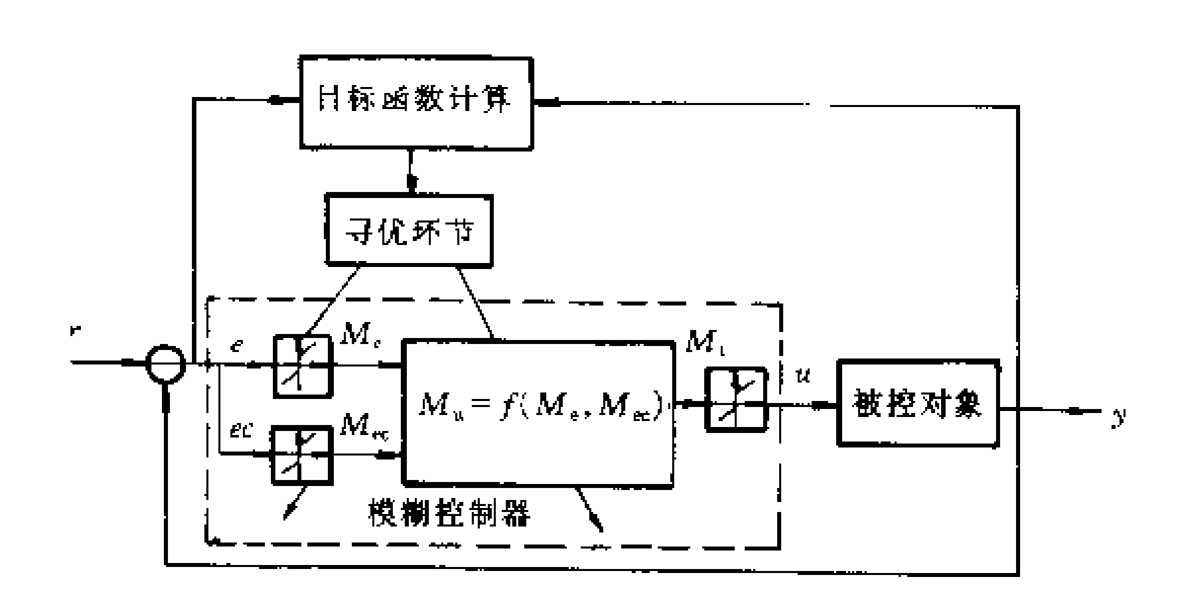


图 4-29 单隶属度函数模糊控制结构图

④ 采用变步长的步进搜索法寻优参数 p 和 α。先寻优参数 p，令 $p_{i+1}=p_i-k_p\Delta J_i \mathrm{sign}(\Delta p_i)$，其中 $\Delta J_i=J_i-J_{i-1}$，$\Delta p_i=p_i-p_{i-1}$，$k_p>0$。寻得 p^* 后，再寻优参数 α，同样地令 $\alpha_{i+1}=\alpha_i-k_\alpha\Delta J_i \mathrm{sign}(\Delta\alpha_i)$，其中 $\Delta\alpha_i=\alpha_i-\alpha_{i-1}$，$k_\alpha>0$。返回执行步骤(3)，直至 J 小于某一规定值。

单隶属度函数模糊控制方法与常规模糊控制器相比，简化了量化、语言变量分档及语言变量值的隶属度函数定义等环节，将诸多具有人为因素的环节合并起来，这样在系统运行过程中更利于调整。它之所以称为模糊控制方法，是因为其中可以包含很多的人为因素，例如 M_e 和 M_{ec}的曲线可以不用 $y=x^p$ 的形式表达，而是在具有操作经验的人绘制的曲线的基础上进行在线调整，控制算式也可根据经验确定。

对于本例所述的单隶属度模糊控制方法，当 $p=1$ 时，即 $M_e=e$；$M_{ec}=\Delta e$；$M_u=u$；$\alpha=0.5$ 时，控制算式变为：

$$u=0.5e+0.5\Delta e$$

这相当于一个 $K_p=0.5$、$K_d=0.5$ 的比例微分控制器。具有在线调整功能的 PD 控制器可以在线地改变 K_p 和 K_d，它对参数空间是线性调整的。而单隶属度函数模糊控制方法的控制规律式等价于

$$u=\alpha e^p+(1-\alpha)\Delta e^p$$

它在线地调整参数 p 和 α，对参数空间是非线性调整的，从这一意义上讲，它是对 PD 控制器的扩展。

3. 仿真研究

在 MATLAB 环境下对该算法进行仿真研究，系统结构如图 4-30 所示。

图 4-30 中的被控对象为$\frac{5}{s^2+5s+6}$，经寻优获得最优值 $p^*=0.898\ 7$；$\alpha^*=0.217\ 2$。这

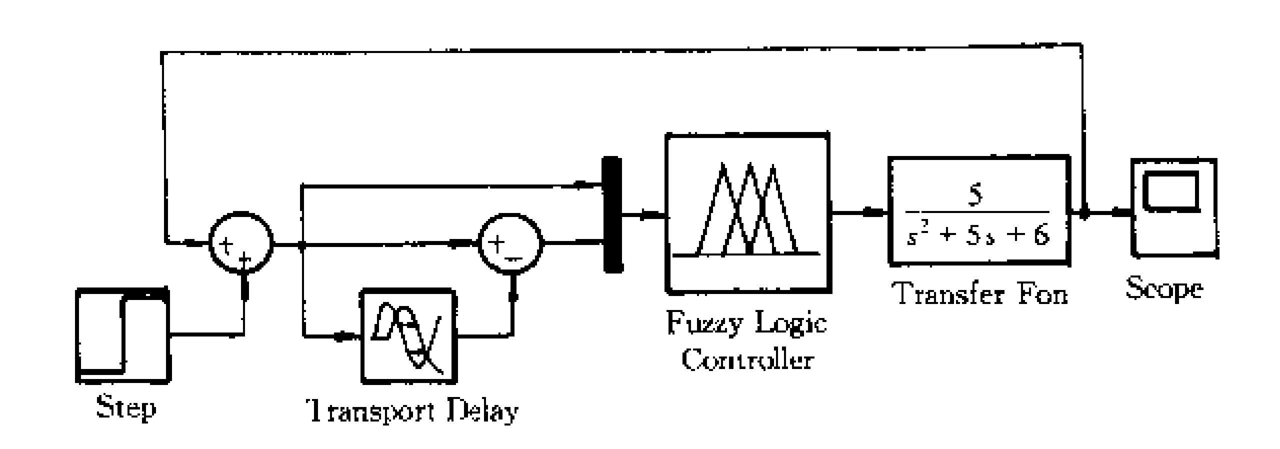


图 4-30　单隶属函数控制仿真系统结构图

两个最优值作为下面三种情况的初始值。

① 改变图 4-30 中对象的时间常数，其传递函数变为$\frac{5}{s^2+4s+3}$，寻优过程见图 4-31。其中 a 为被控对象未改变时的输出响应曲线；b，c 为寻优过程；d 为最终的寻优结果。寻得 $p^*=0.9487$；$\alpha^*=0.1074$。

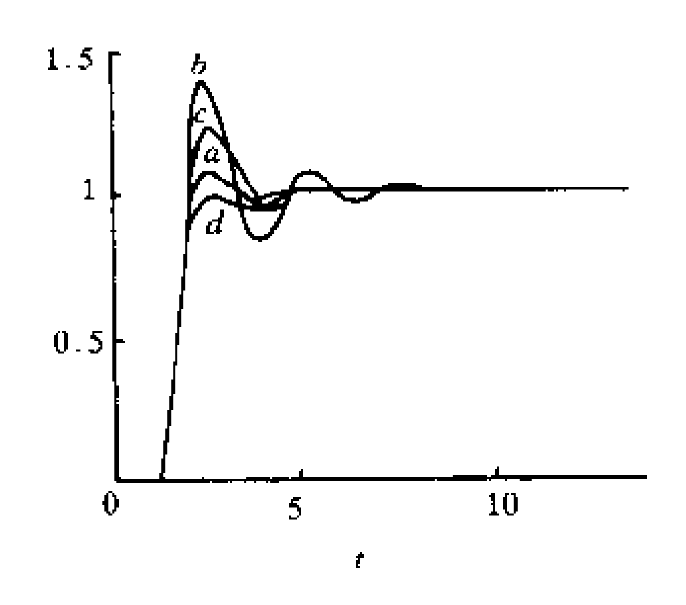


图 4-31　改变对象时间常数的寻优过程

② 改变图 4-30 中对象的放大系数，其传递函数变为$\frac{10}{s^2+5s+6}$，寻优过程见图 4-32。其中 a 为被控对象未改变时的输出响应曲线，b，c 为寻优过程，d 为最终的寻优结果，寻得 $p^*=0.9387$；$\alpha^*=0.1244$。

③ 改变图 4-30 中对象的阶次，其传递函数变为$\frac{5}{s^2+5s+6}\times\frac{1}{s+2}$，寻优过程见图 4-33。其中 a 为对象未改变时的输出响应曲线，b，c，d 为寻优过程，e 为最终结果，寻得 $p^*=0.9398$；$\alpha^*=0.1760$。

对单隶属度函数模糊控制方法的初步仿真研究表明这一算法是简单而有效的。对算法中

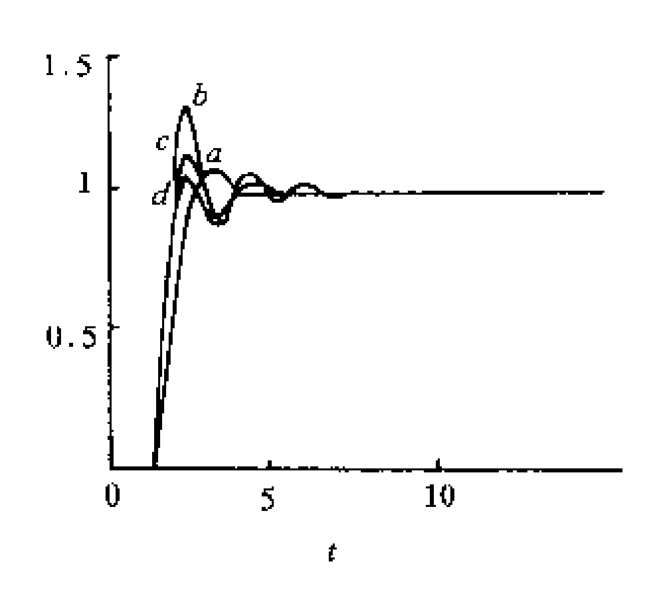


图 4-32　改变对象放大系数的寻优过程

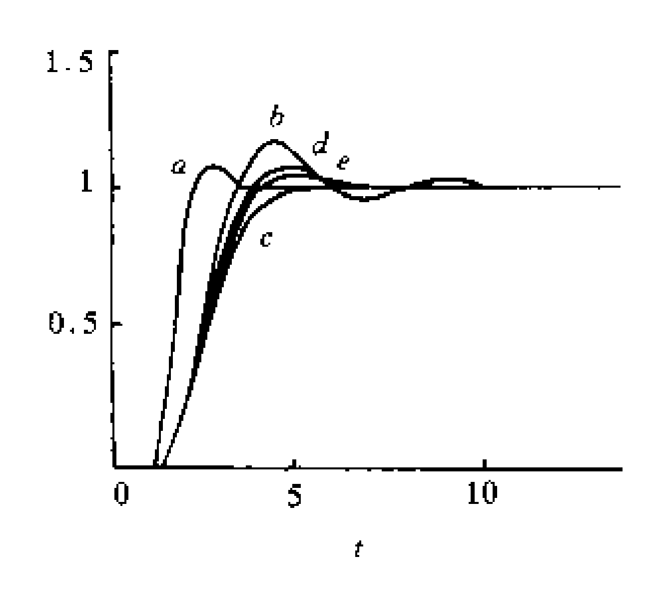


图 4-33　改变对象阶次的寻优过程

的很多细节问题和对这一算法的扩展及该模糊控制器的仪表化等诸多问题尚在探讨和研制之中。

4.5　水温模糊控制系统

温度控制方法基本上有两种：一种是采用位式调节，这种方法控制线路简单，可靠性较高，但控制精度低；另一种是采用 PID 控制，这种方法响应迅速、稳态误差小，但当系统内部参数发生变化或受到外界干扰时参数整定困难。为了满足实验室催化槽水温控制的要求，设计一种带修正因子和寻优参数的水温模糊控制系统，既保留了前两种方案的优点，又能克服它们的缺点，尤其是该系统具有较强的适应系统内部参数变化和克服外界干扰的能力。

1. 系统描述

为了保证某大型实验的顺利进行，要求容积为5L的催化槽中水温可在45℃～85℃之间任意设定，且在长时间的保温阶段温度波动≤±1℃，稳态精度<0.3℃。为了满足这一要求，设计了带修正因子和寻优参数的水温模糊控制器。

控制规则是模糊控制器的核心部分，它决定了模糊控制器的有效性和实用性。具有一成不变的控制规则的控制器的适应性显然较差，故采用一种带修正因子的模糊控制算法：

$$U=\mathrm{INT}[\alpha E+(1-\alpha)EC] \qquad \alpha\in[0,1]$$

式中，α为加权系数；INT[·]表示对"·"取整，α值的大小直接反映了控制量U对误差E及误差变化率EC的加权程度，其值可根据误差E的大小在线自动调整。设误差E和误差变化率EC的论域为

$$\{E\}=\{EC\}=\{-N,-N+1,\ldots,-2,-1,0,1,2,\ldots,N\}$$

则在整个论域范围内带在线自调整参数的模糊控制规则可表达为

$$\begin{cases}U=\mathrm{INT}[\alpha E+(1-\alpha)EC]\\ \alpha=\alpha_0+(1-\alpha_0)\dfrac{|E|}{N}\end{cases}$$

式中，$\alpha\in[\alpha_0,1]$，α_0一般取0.3左右。上述控制规则的特点是，调整参数α在α_0至1之间随误差绝对值$|E|$的大小而变化，因N是量化等级，故α有$N+1$个可能的取值。当$\alpha=\alpha_0$时，α为最小，对EC的权重最大；当$|E|=|E|_{max}$时，$\alpha=1$，使E权重最大以尽快消除误差。

图4－34为带修正因子的规则自调整模糊控制器。

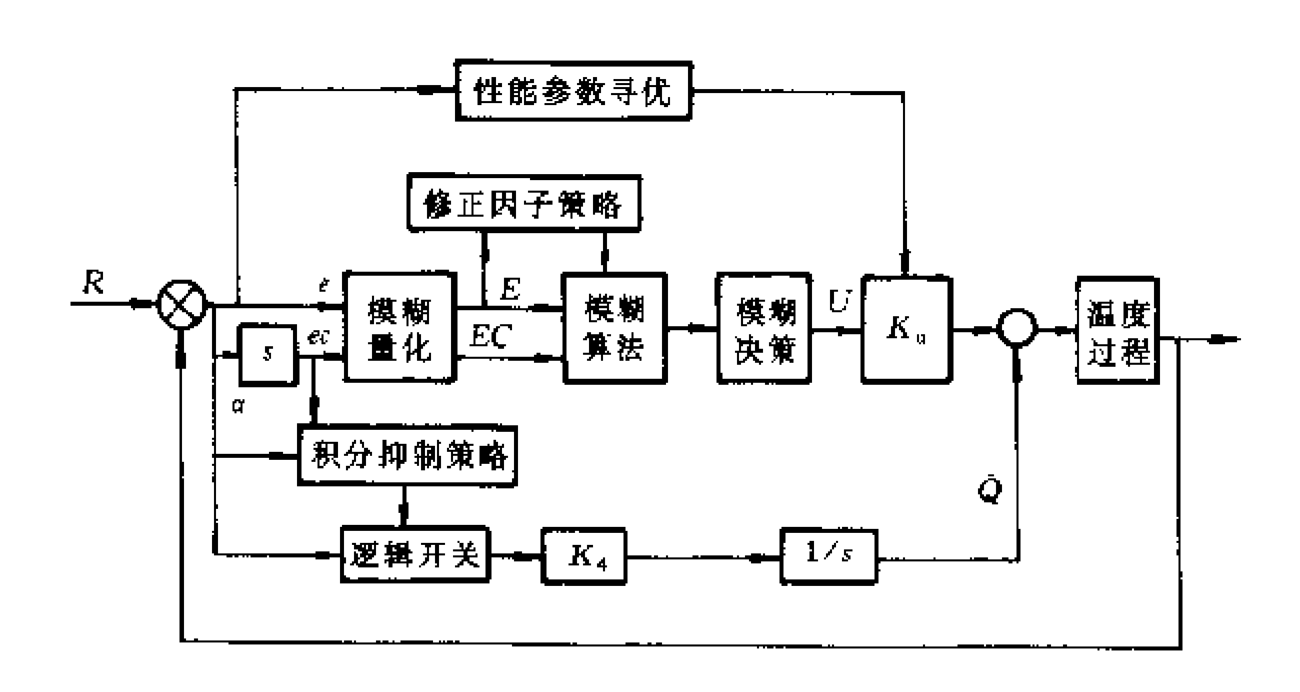


图4－34　带修正因子的规则自调整模糊控制器

其中积分抑制策略是为提高稳态精度、增强适应性和进一步抑制超调而设，其策略为

if $|E|<M_4$ and $|EC|<N_4$ and $EC \cdot E \leqslant 0$

then $Q(k)=Q(k-1)+K_4 \cdot e(k)$

else $Q(k)=Q(k-1)$

其中，M_4，N_4 分别为给定的阈值；K_4 为比例系数，模糊量化部分采用归一模糊量化，这样就无需再设置和调整量化因子 K_e 和 K_{ec}，这时模糊控制器所需调整的参数只剩下 K_u 一个，可利用自寻优技术在线调整 K_u，使系统工作在最佳状态。在此选用 ITAE 准则作为目标函数，即使 $J=\int t\mid e(t)\mid \mathrm{d}t$ 取最小值。

采用带修正因子和优化参数的模糊控制器，可使模糊控制规则实现自调整、自完善，适用于被控对象的特性不太明确或过于复杂而无法用精确数学模型来描述的系统。这种方法不仅可保证系统输出响应快、调节时间短、超调量小、控制精度高，而且还具有较强的适应系统内部参数变化和抵抗外部扰动的能力。很多情况下温度对象的数学模型难以建立，易受外部扰动的影响，且被控性能要求较高，因此适合于用这种模糊控制器加以控制。

2. 仿真结果

下面是对在已知系统参数和系统参数可变两种情况下，应用模糊控制和 PID 控制进行控制的结果的比较。

① 已知系统参数的 PID 控制器最优设计。如图 4－35，假定系统对象已知，在 PID 参数的优化设计中采用ITAE准则：$J=\int_0^{t_s} t\mid e(t)\mid \mathrm{d}t$，得系统PID控制器的最优参数为[1.7，1.4，0.3]。图 4－35 中 $G(s)$和 $G_1(s)$分别为

$$G(s)=\frac{1}{s+0.5}$$

$$G_1(s)=\frac{4.228}{(s^2+1.645+8.456)}$$

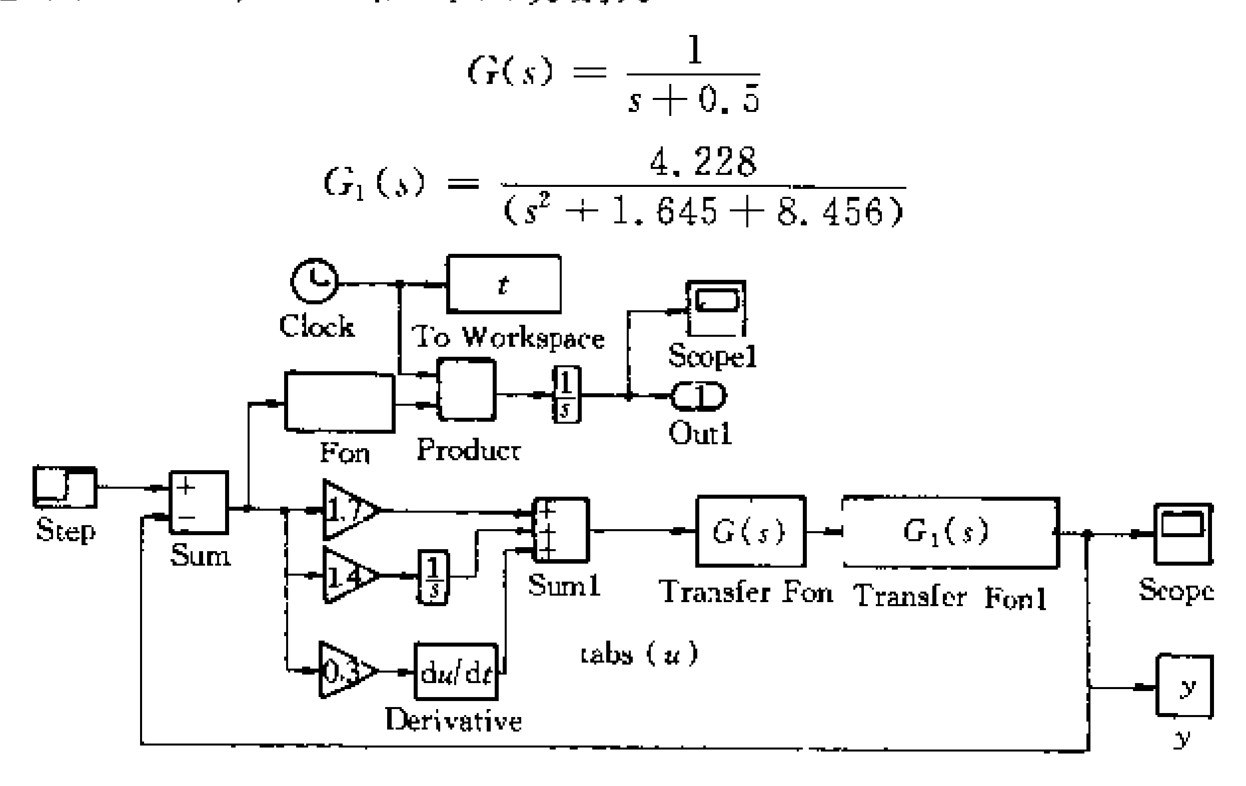


图 4－35　水温 PID 控制仿真系统

② 模糊控制系统设计。如图 4－36 为具有模糊增益调整的 PID 控制器设计，假定 K_p，K_d，K_i 分别表示比例增益、微分增益、积分增益。参数由当前误差和它的一阶差分根据一组模糊规则决定：

If $e(k)$是 A_i 和 $\Delta e(k)$是 B_i，then K_p 是 C_i，K_d 是 D_i 和 $\alpha=\alpha_i(i=1,2,\dots,m)$

这里 A_i，B_i，C_i 和 D_i 是在相应支集上的模糊集合。

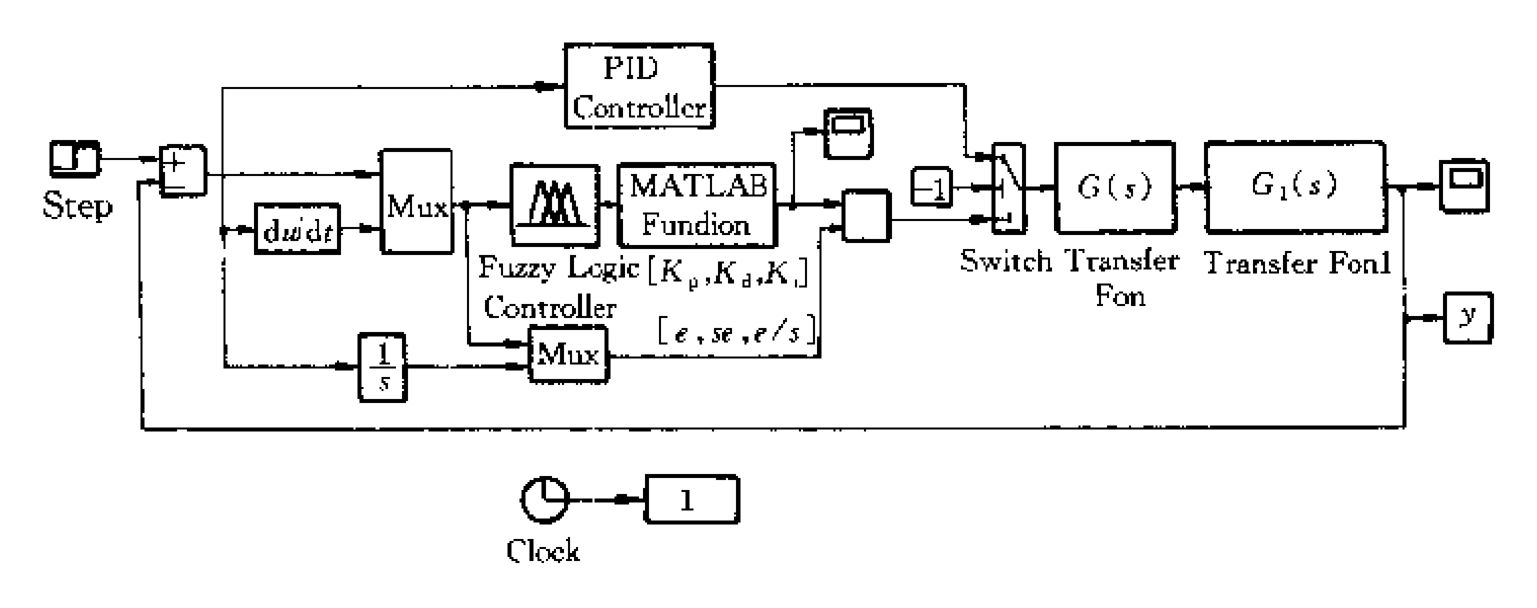


图 4－36　水温模糊控制仿真系统 1

③ 两者仿真效果比较结果如图 4－37 所示。

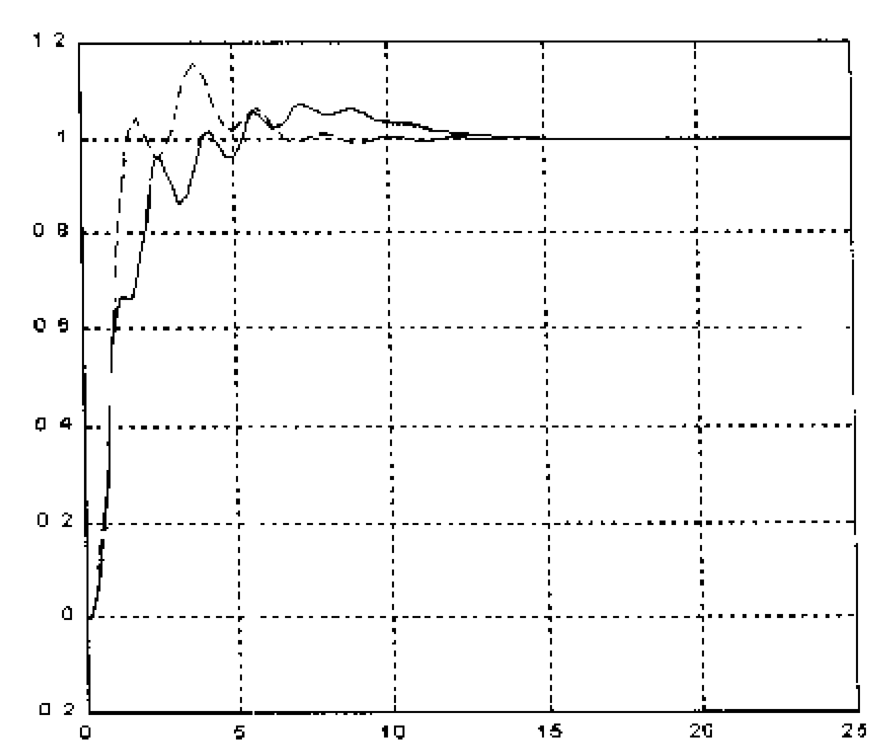


图 4－37　三阶系统性能比较（虚线——常规 PID 控制；实线　模糊控制）

④ 当控制对象的参数改变时两种控制方法的比较。其系统仿真结构和仿真结果分别如图 4－38 和图 4－39 所示。图 4－38 中的 $G(s)$为

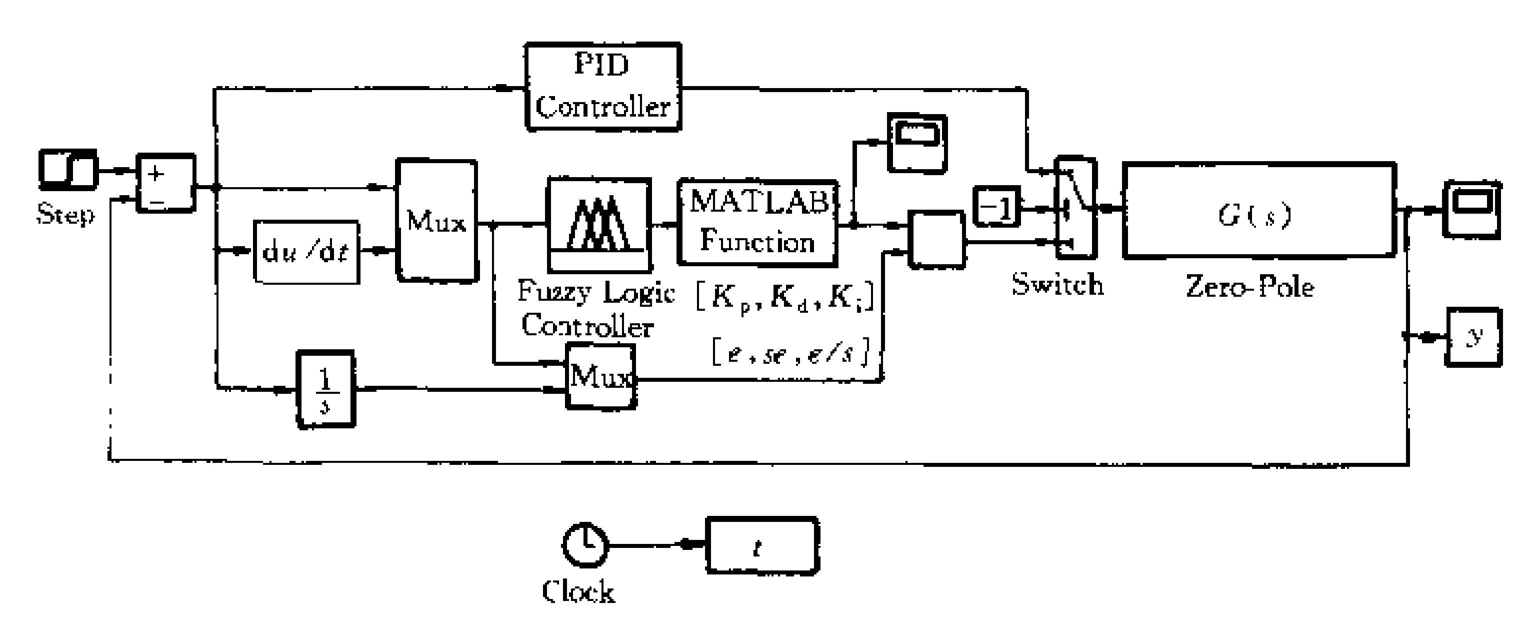


图 4-38　水温模糊控制仿真系统 2

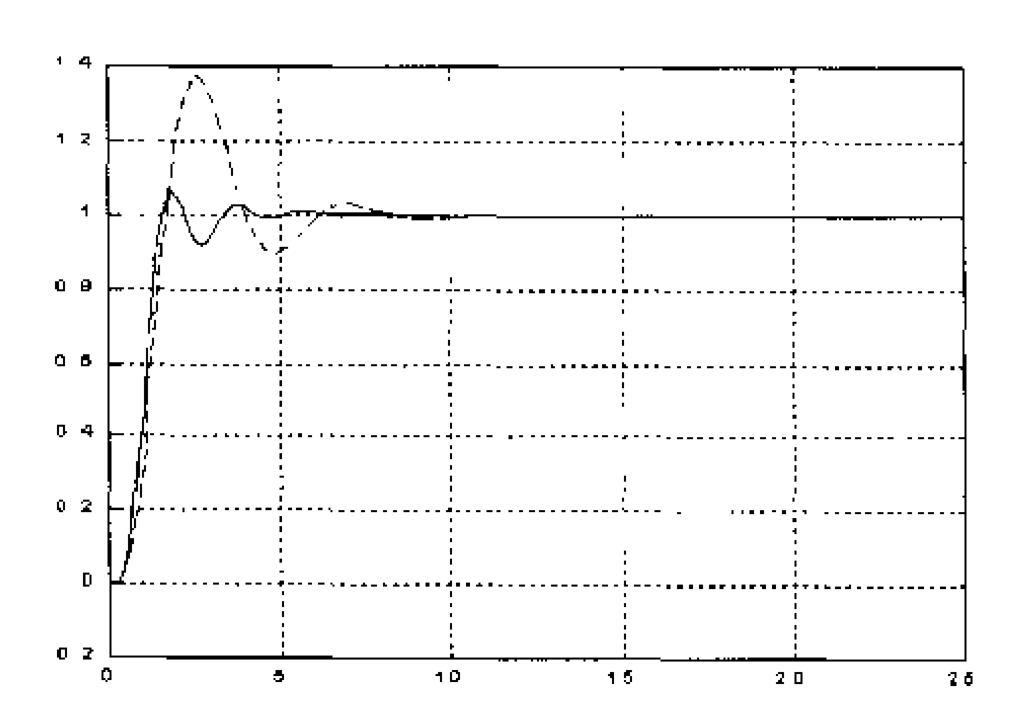


图 4-39　四阶系统性能比较(虚线——常规 PID 控制;实线　　模糊控制)

$$G(s) = \frac{27}{(s+1)(s+3)^3}$$

⑤ 仿真结果分析。PID 控制器结构简单、性能良好,特别适用于被控对象参数固定,非线性不很严重的情况。但是在过程参数变化比较大的时候,PID 控制器就难以取得良好的性能。因而就需要对 PID 参数进行在线的调整。模糊增益调整 PID 控制器就是利用模糊规则和推理来对 PID 参数进行在线调整,效果良好。

4.6　电液伺服模糊控制系统

模糊控制系统动态性能好，但稳态性能较差，且很难使两种性能指标都达到理想要求。因此，研究者提出了模糊-线性复合控制器的设计思想。目前普遍采用的模糊-线性复合控制器有四种型式：引入线性前馈的复合控制；模糊-线性双模控制；在线性控制的误差通道并联模糊控制器；模糊控制器与线性控制器并联。模糊-线性复合控制发挥了模糊控制和线性控制的优点，使设计的系统取得了较好的动态和稳态指标。但是，模糊-线性复合控制同时也存在一些问题：线性前馈复合控制的系统性能对参数变化比较敏感；模糊-线性双模控制存在开关切换问题等。在线性控制的误差通道并联模糊控制器系统和模糊控制器与线性控制器并联系统尽管得到了较好的效果，但并不能从根本上解决模糊控制器稳态性能和动态性能之间的矛盾问题。

为此，研究者从修改模糊控制规则入手，提出了一种新的模糊控制器设计方法：由控制对象的动态响应特性和模糊控制产生极限环振荡的条件为依据，在模糊控制规则表中确定一准稳态区，在准稳态区内，采用确定性系统最优控制方法，从而使控制系统不仅有很快的动态响应速度，且有很好的稳态性能。

1. 研究分析

普通的模糊控制器具有 PD 调节特性，即控制器通过由系统的偏差 E 及偏差的变化率 EC 组成的二维控制表来决定控制量 U 的变化规律。当系统偏差不在＜0＞等级内时，模糊控制器发出控制量 U，力图使偏差回到＜0＞等级。一旦偏差回到＜0＞等级，系统将处于瞬时开环状态，此时输出为零，但由于惯性负载或负载的影响，系统很快又会跳离＜0＞等级。如此反复，便会形成极限环振荡。

模糊-线性复合控制的主要思想是：系统偏差 E 进入＜0＞等级后，系统的瞬时输出为零，从而自动使系统切换到线性控制器状态。但是，这里需注意的是：当系统进入＜0＞等级后，不可能稳定在＜0＞等级。由于系统惯性的影响，系统很快又会偏离＜0＞等级，依然会出现极限环振荡。如果要避免极限环振荡，就需要将＜0＞等级调到一定范围宽度，即调整参数 U 的分档间隔参数。但是由模糊控制理论可知，在选择档极宽度的参数时，偏差较大的区域应选用低分辨率的模糊集（曲线缓），偏差较小的区域应选用较高分辨率的模糊集（曲线较尖），偏差接近于零时应选用高分辨率的模糊集（曲线很尖）。因此，分档间隔参数的变化范围是有一定限度的，并不能无限制地调宽，而且分档间隔参数的选取与控制对象的性能指标间的关系也很难确定。这就使模糊-线性复合控制的应用受到一定限制。

在设计模糊控制器时，通常要做精确量的模糊化、模糊控制规则的选择以及输出信息的模糊判决三方面的工作，因此，使得最后制定的模糊控制表包含一定程度的非线性。这势必会对

闭环系统的响应、动态品质产生一定的影响。通过傅立叶变换，应用描述函数法分析了模糊控制系统的非线性特性和持续振荡等问题，表明极限环振荡的实质是由于被控对象的幅相特性曲线与模糊控制器描述函数 $f(E,EC)$ 的负倒频率特性曲线相交且有稳定交点，闭环系统将在对应于该交点的幅值和频率下做等幅振荡。如果开环幅相特性曲线包围 $f(E,EC)$ 的负倒频曲线或其交点是非稳定的，则表明系统如按现行的模糊控制表进行控制，将不可能稳定工作。因此，模糊控制器描述函数 $f(E,EC)$ 的负倒频特性与模糊控制表的制定原则有直接关系。

综合考虑上述原因，研究者提出一种新型的模糊控制器设计方法，介绍如下。

2. 模糊控制器的设计

模糊规则的选择是设计模糊控制器的核心。通常人们认为模糊规则的选择主要依赖于设计者的经验，但是现在可以从控制规则状态表的物理意义和模糊控制器的描述函数 $f(E,EC)$ 两方面来考虑模糊控制规则的选取问题。

用控制表的物理意义来指导模糊规则的选取是指：在系统调试时根据系统的现实响应与对其要求，有目的、有针对性地对控制规则进行修改。如当系统的偏差 E 为 PB，EC 也为 PB 时，说明系统正处于响应的上升阶段，因而系统的控制量 U 也应该是 PB，以便使系统的偏差迅速减小。而当系统的偏差 E 为 PS，EC 为 NS 时，则说明系统正处于稳态区附近，所以控制量 U 可取为 PO 或 NO，以保证系统稳定在稳态区附近。

用模糊控制器的描述函数来指导模糊规则的选取是指，通过描述函数的负倒频曲线和被控对象的幅频曲线的稳定交点（可能不止一个）来决定模糊控制表的分区问题。这里的分区问题不是指正大、正小的划分问题，而是指为了避免系统的自振荡或极限环振荡而划分的区。离原点最远的点对应于系统大偏差时的持续振荡，离原点最近的点对应于系统小偏差时的持续振荡（极限环振荡）。为了消除极限环振荡，可在模糊控制规则表内划分一个准稳态区，该区的起点表现在 $f(E,EC)$ 的负倒频曲线中，应位于稍大于产生极限环振荡的交点处。换句话说，就是通过交点来决定偏差及偏差范围各档的取值范围，即各档隶属函数曲线的急缓程度。在准稳态区内，模糊控制规则可按确定性系统的控制思想进行设计，以使系统的稳态性能满足设计要求。

3. 仿真实验

图 4-40 给出一个电液伺服系统的框图。该系统的负载为一土试件。在试验过程中，土试件刚度有很大的变化，且具有一定的不确定性，这使得系统在常规的控制方法下不能得到很好的动静态特性。这里借助 MATLAB 的模糊推理系统工具箱，按照上述的模糊控制器设计方法来调节系统的动静态特性。希望通过模糊控制能够补偿负载特性（柔性）对系统的性能的影响，并减小系统的稳态误差。

模糊推理工具箱有一个独立的编辑环境 FIS Editor，它将模糊控制器的整个设计过程融为一体。通过 FIS Editor 可进行模糊推理方法和模糊判决方法的选择及隶属函数的编辑、输

入模糊控制规则等操作。现用模糊推理工具箱分别如图 4－40(a)和(b)所示。系统仿真结果见图4 41。

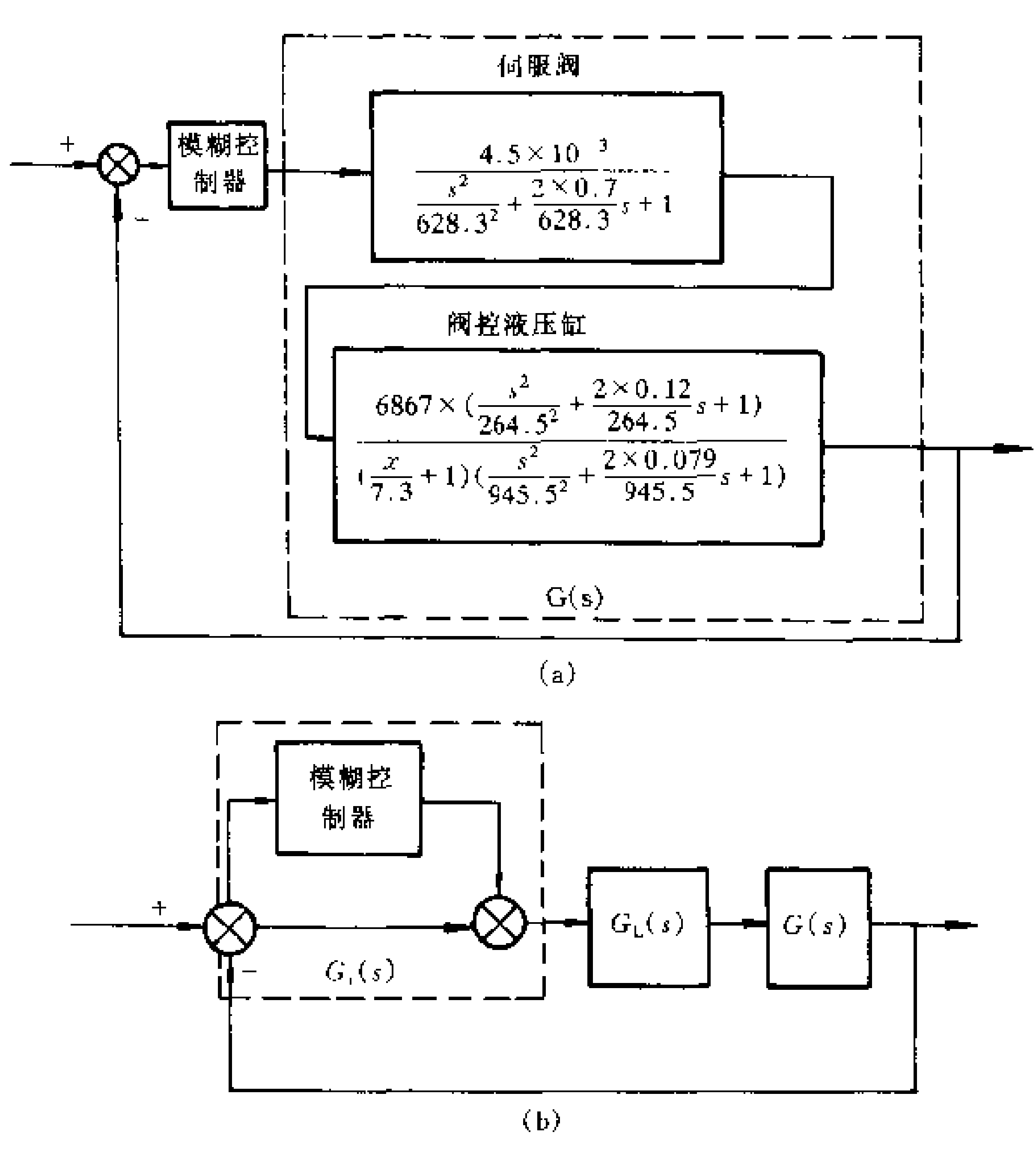


图 4－40　电液伺服系统

可见，按本例所述方法适当选取模糊控制规则，不难获得高的稳态精度和快的动态响应。模糊-线性复合控制在这里应用的效果不是很好的主要原因是：在模糊-线性复合控制中$G_c(s)$并不是单纯的模糊控制，而是系统偏差与模糊控制器的并联，这使得模糊控制器对系统的控制作用并不是十分的明显。

4. 结论

综上所述，采用本例所介绍的模糊控制器设计方法，可得到较快的动态响应，较小的超调量，且比以往的模糊控制器有较高的稳态精度。另外，在此所述方法是通过修改模糊控制规则，从模糊控制器本身入手来解决稳态精度和动态响应速度之间的矛盾的，这与其它模糊控制器的设计方法是有本质区别的。

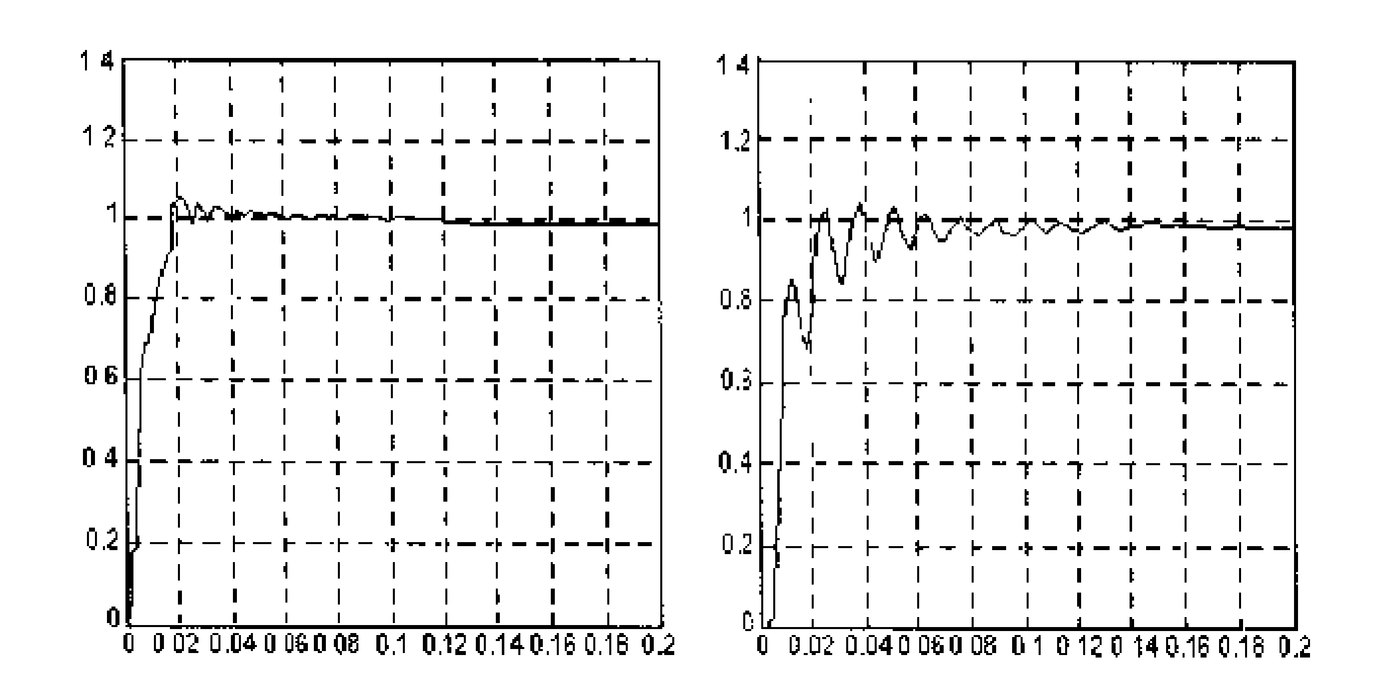

0 0.2 0.4 0.6 0.8 1 1.2 1.4
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2
0 0.2 0.4 0.6 0.8 1 1.2 1.4
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2

图 4－41　系统仿真结果

附录 模糊逻辑工具箱函数详解

A. 图形界面工具

函　数	功　能
anfisedit	ANFIS 图形界面编辑器
fuzzy	基本模糊推理系统编辑器
mfedit	隶属函数编辑器
ruleedit	模糊规则编辑器及其分析
ruleview	模糊规则观察器及模糊推理图形
surfview	输出曲面观察器

1. anfisedit

(1) 功能

Sugeno 型 FIS 训练程序。

(2) 使用格式

```
[fismat,error1,stepsize]=anfis(trnData)
[fismat,error1,stepsize]=anfis(trnData,fismat)
[fismat1,error1,stepsize]=...
anfis(trnData,fismat,trnOpt,dispOpt)
[fismat1,error1,stepsize,fismat2,error2]=...
anfis(trnData,trnOpt,dispOpt,chkData)
[fismat1,error1,stepsize,fismat2,error2]=...
```

anfis(trnData,trnOpt,dispOpt,chkData,optMethod)

(3) 说明

这是 Sugeno 型模糊推理系统主要的训练程序。anfis 使用一种混合训练算法辨识 Sugeno 型模糊推理系统的参数。在训练 FIS 隶属函数参数中,它应用了反向传播下降梯度法与最小二乘法相结合的方法,去模拟给定的训练数据集。在模型确认中,可用一些可选参数调用 anfis。随着这些选择产生的模型确认的类型是对模型完全匹配的检验,其参数是一个数据集合,称为检验数据集。

anfis 的各输入参数:

① trnData:训练数据集名称。它是一个矩阵。除末列为包含输出数据的单个矢量外,其余各列中包含了所有的输入数据。

② fismat:模糊推理系统名。在训练中,它向 anfis 提供一个隶属函数初始集。如果没有该选项,anfis 将在训练中使用 genfis1 以实现默认的初始 FIS。当只用一个参数调用 anfis 时,默认的 FIS 有两个高斯型隶属函数。如果 fismat 作为一个单独的数字(或一个矢量)被提供给 anfis,则该数字被用作隶属函数的个数(或者,当对每个输入的隶属函数个数不同时,矢量中的每一元素分别是与每个输入相关的不同隶属函数的个数)。在这种情况下,在训练开始之前,anfis 的两个参数就被传递给 genfis1 以产生一个有效的 FIS 结构。

③ trnOpt:训练选项矢量。只要任何训练选项的输入为 NaN,则强制使用默认选项。TrnOpt 向量中的各项分别为:

trnOpt(1):训练时段数(默认值:10)。

trnOpt(2):训练误差目标值(默认值:0)。

trnOpt(3):初始步长(默认值: 0.01)。

trnOpt(4):步长减少速度(默认值: 0.9)。

trnOpt(5):步长增加速度 (默认值: 1.1)。

④ dispOpt:显示选项矢量。它指定训练时在 MATLAB 命令窗口中显示的信息。对任何显示选项的默认值均为 1,即表示显示所有的响应信息。当设为 0 时,则该项响应信息不在屏幕上显示。如果输入的显示选项为 NaN,则强制使用缺省选项。DispOpt 向量中的各项分别为:

dispOpt(1):ANFIS 信息,例如输入输出隶属函数个数等(缺省值:1)。

dispOpt(2):误差(缺省值:1)。

dispOpt(3):在每个参数更新时的步长(缺省值:1)。

dispOpt(4):最终结果(缺省值:1)。

⑤ chkData:在完全匹配模型确认中的可选检验数据集名称。该数据集为一个矩阵,其格式与训练数据集相同。

⑥ optMethod：在隶属函数参数训练中使用的可选优化方法。设为1时，使用混合法；设为0时，使用反向传播算法。默认方式是混合法，即最小二乘法与反向传播算法相结合的方法。当该参数的输入为任何非0的值时，都将调用默认方式。

在训练中，只要达到指定的训练时长或训练误差目标，则训练停止。

注意： 当用两个或更多个参数调用 anfis 时，如果对其它的可选参数项输入为 NaN 或空矩阵，则它们都将以默认方式启动。在文件 anfis.m 中可以直接修改 anfis 的默认值。如果用户不想指定它们，但又想指定后续的参数，例如应用 anfis 的检验数据选项，则不论 NaN 还是空矩阵都可用来对变量"占位"，以便指定后续参数。

anfis 的各输出变量如下：

① fismat1：FIS 结构，其参数根据训练误差标准的最小值设置。

② error1 或 error2：分别表示训练数据误差信号和检验数据误差信号的均方根误差。

③ Stepsize：步长数组，如果误差测量连续两次一增一减，则步长减小（用与步长减小速度相应的训练选项元素与其相乘）；如果误差测量连续四次减小，则步长增加（同步长增加速度与其相乘）。

④ fismat2：FIS 结构，其参数根据最小检验误差标准设置。

参见：

genfis1，anfis

2. fuzzy

(1) 功能

调用基本 FIS 编辑器。

(2) 使用格式

fuzzy

fuzzy(fismat)

(3) 说明

利用该图形化工具，可以编辑模糊推理系统的最高级特征，例如输入输出变量的个数，所使用的反模糊化方法等。要更详细地了解 fuzzy 图形化工具的使用方法，请参阅"FIS 编辑器"。

FIS 编辑器中给出了任一模糊逻辑推理系统的高级特征。它允许用户调用其它编辑器对 FIS 进行操作。在与模糊系统的交互作用中，FIS 编辑器界面极具灵活性，使用户可以方便地访问所有的其它编辑器。

参见：

mfedit，ruleedit，ruleview，surfview，anfisedit

3. mfedit

(1) 功能

隶属函数编辑器。

(2) 使用格式

mfedit('a')

mfedit(a)

mfedit

(3) 说明

mfedit('a')产生一个隶属函数编辑器,可以用它检查并修改存储在文件 a. fis 中的模糊推理系统的所有隶属函数。

mfedit(a)是对 MATLAB 工作区变量的操作。对一个 FIS 结构,单独的 a. mfedit 打开隶属函数编辑器但不载入 FIS。

利用隶属函数编辑器可以修改每一隶属函数的名称、类型和参数,并完全可以自定义特定的隶属函数。但用户也可以从工具箱提供的 11 个内置隶属函数中任选其一。要得到更多的使用 mfedit 的信息,请参阅"隶属函数编辑器"一节。

要在图形区中显示与某一变量相关的隶属函数,请首先在图形的左上部(标题 FIS Variables 下面)选择变量的图标。要选择隶属函数,请在隶属函数标号上单击。

参见:

fuzzy, ruleedit, ruleview, surfview

4. ruleedit

(1) 功能

模糊规则编辑器及其分析程序。

(2) 使用格式

ruleedit('a')

ruleedit(a)

(3) 说明

当模糊规则编辑器被 ruleedit('a')调用时,它用于修改存储在文件 a. fis 中 FIS 结构的模糊规则,还可被模糊推理系统用于检验规则。

要用这个编辑器建立规则,必须首先定义想在 FIS 编辑器使用的所有输入和输出变量。对输入和输出变量、连接以及权值项使用列表框和复选框选项,就可以建立规则。

要得到对 ruleedit 更多的使用信息,请参阅"规则编辑器"一节。

参见:

addrule,fuzzy,mfedit,parsrule,ruleview,showrule,surfview,view

5. ruleview

(1) 功能

模糊规则观察器及模糊推理图。

(2) 使用格式

ruleview('a')

(3) 说明

使用 ruleview('a')可调用模糊规则观察器,以描述存储在文件 a.fis 中 FIS 的模糊推理图。模糊规则观察器用于观察从开始到结束的整个关联过程。用户可以移动相应于输入的线条指针,然后观察系统的重新调整并计算出新的输出。

要得到更多的使用 ruleview 的信息,请参阅“规则观察器”一节。

参见:

fuzzy,mfedit,ruleedit,surfview。

6. surfview

(1) 功能

输出曲面观察器。

(2) 使用格式

surfview('a')

(3) 说明

使用 surfview('a')可调用输出曲面观察器。输出曲面观察器是一个图形化工具,可以用它观察存储在文件 a.fis 中的模糊推理系统对任何单输入或二输入的输出曲面。

由于输出曲面观察器不以任何方式改变模糊系统及其相关的 FIS 结构,因此,它是一个只读编辑器。

使用列表框,可以选择任何输入变量构成两个输入坐标轴(X 轴和 Y 轴),同时可以选择任何输出变量构成输出坐标轴(Z 轴)。

选中 Option 菜单下的 Always Evaluate,输出曲面观察器将即时执行计算并绘制输出曲面。

按住坐标轴并拖动鼠标,就可以从不同角度观察输出曲面。

如果模糊推理系统的输入个数大于两个,在指定输入时必须将其它未指定的输入设为常量。

要得到使用输出曲面观察器更多的信息,请参阅“输出曲面观察器”。

参见:

anfisedit,fuzzy,gensurf,mfedit,ruleedit,ruleview

B. 隶属函数

函　数	功　能
dsigmf	两个 sigmoid 函数之差构成的隶属函数
gauss2mf	双边高斯曲线隶属函数
gaussmf	高斯曲线隶属函数
gbellmf	一般钟形隶属函数
pimf	π 曲线形隶属函数
psigmf	两个 sigmoid 隶属函数之积
smf	S 曲线形隶属函数
sigmf	Sigmoid 曲线形隶属函数
trapmf	梯形隶属函数
trimf	三角形隶属函数
zmf	Z 形隶属函数

1. dsigmf

(1) 功能

模糊逻辑工具箱内置函数，它由两个 sigmoidal 隶属函数之差建立。

(2) 使用格式

y=dsigmf(x,[a1　c1　a2　c2])

(3) 说明

sigmoidal 隶属函数由两个参数 a, c 决定，并由下式给出

$$f(x;a,c)=\frac{1}{1+e^{-a(x-c)}}$$

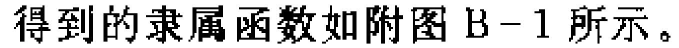
隶属函数 dsigmf 由四个参数 a_1, c_1, a_2, c_2 决定，它是两个 sigmoidal 隶属函数之差：

$$f_1(x;a_1,c_1)-f_2(x;a_2,c_2)$$

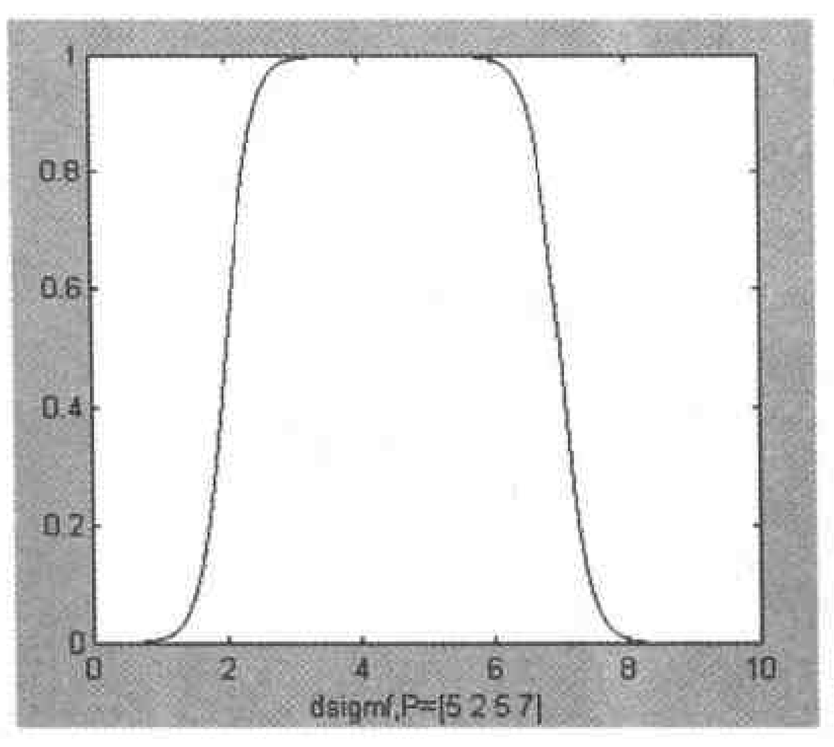
在 dsigmf 中，四个参数的顺序为：[a1 c1 a2 c2]。

例如：

```
x-0:0.1:10;
y=dsigmf(x,[5 2 5 7]);
plot(x,y)
xlabel('dsigmf,P=[5 2 5 7]')
```



得到的隶属函数如附图 B－1 所示。



附图 B－1　dsigmf，两个 sigmoid 函数之差构成的隶属函数

参见：

gaussmf，gauss2mf，gbellmf，evalmf，mf2mf，pimf，psigmf，sigmf，smf，trapmf，trimf，zmf

2. gauss2mf

（1）功能

高斯组合隶属函数。

（2）使用格式

y＝gauss2mf(x,[sig1 c1 sig2 c2])

（3）说明

高斯函数由两个参数 σ 和 c 决定，并由下式给出

$$f(x;\sigma,c)=e^{-\frac{(x-c)^2}{2\sigma^2}}$$

函数 gauss2mf 由两个高斯函数组合而成。第一个高斯函数的参数为 sig1 和 c1，它决定 gauss2mf 曲线左边的形状；第二个高斯函数的参数为 sig2 和 c2，它决定 gauss2mf 曲线右边的形状。

如果 c1＜c2，则 gauss2mf 函数的最大值可为 1，否则其最大值将小于 1。

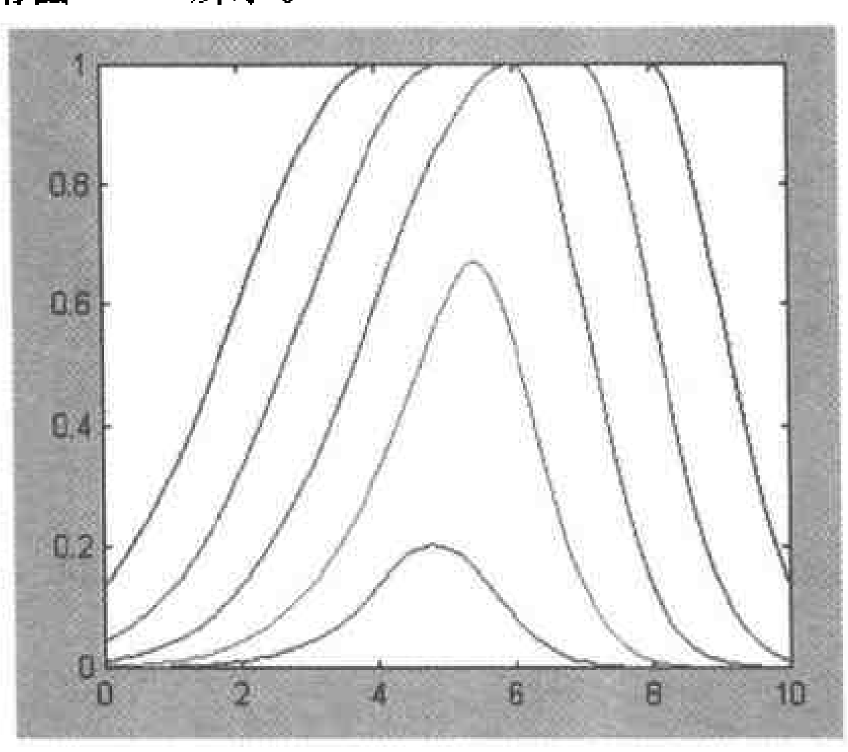
在函数 gauss2mf 中，各参数的顺序为：[sig1，c1，sig2，c2]。

例如：

```
x=(0:0.1:10);
y1=gauss2mf(x,[2 4 1 8]);
```

```
y2=gauss2mf(x,[2 5 1 7]);
y3=gauss2mf(x,[2 6 1 6]);
y4=gauss2mf(x,[2 7 1 5]);
y5=gauss2mf(x,[2 8 1 4]);
plot(x,[y1 y2 y3 y4 y5]);
set(gcf,'name','gauss2mf','numbertitle','off');
```

得到的隶属函数如附图 B－2 所示。



附图 B－2　gauss2mf,高斯组合隶属函数

参见：

dsigmf,gauss2mf,gbellmf,evalmf,mf2mf,pimf,psigmf,sigmf,smf,trapmf,trimf,zmf

3. gaussmf

(1) 功能

模糊逻辑工具箱内置的高斯隶属函数。

(2) 使用格式

y=gaussmf(x,[sig c])

(3) 说明

对称高斯函数由两个参数 σ 和 c 决定,并由下式给出

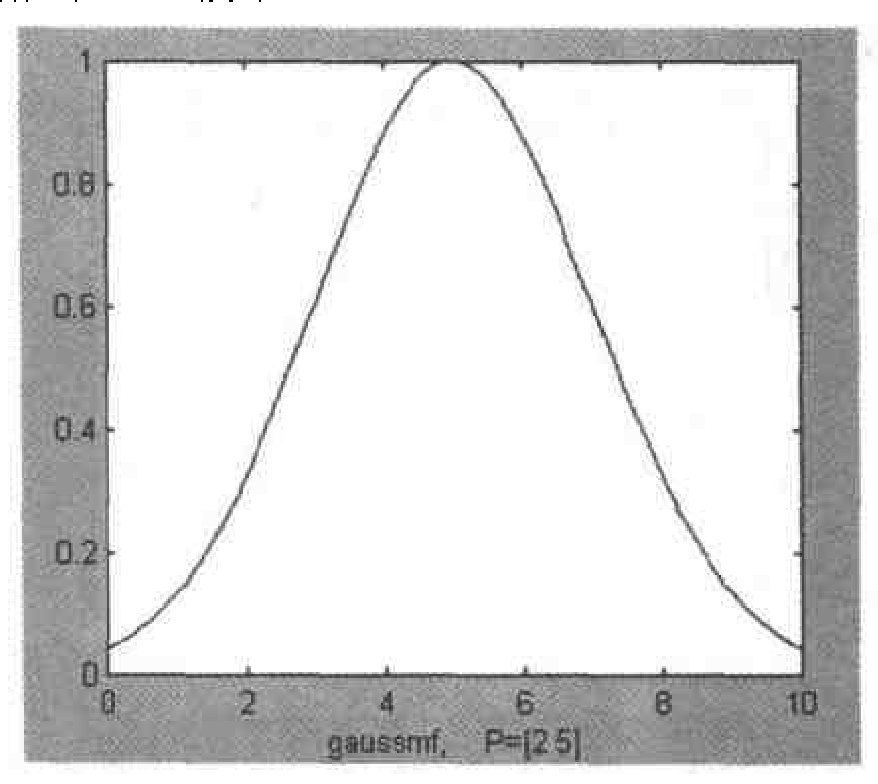
$$f(x;\sigma,c)=e^{\frac{(x-c)^2}{2\sigma^2}}$$

在 gaussmf 中,参数的顺序为:[sig c]。

例如:

```
x=0:0.1:10;
y—gaussmf(x,[2 5]);
plot(x,y)
xlabel('gaussmf, P—[2 5]')
```

得到的隶属函数如附图 B-3 所示。



附图 B-3　gaussmf,基本高斯隶属函数

参见:

dsigmf,gaussmf,gbellmf,evalmf,mf2mf,pimf,psigmf,sigmf,smf,trapmf,trimf,zmf

4. gbellmf

(1) 功能

模糊逻辑工具箱内置的广义钟形隶属函数。

(2) 使用格式

```
y=gbellmf(x,params)
```

(3) 说明

广义钟形函数由三个参数 a,b 和 c 决定,并由下式给出

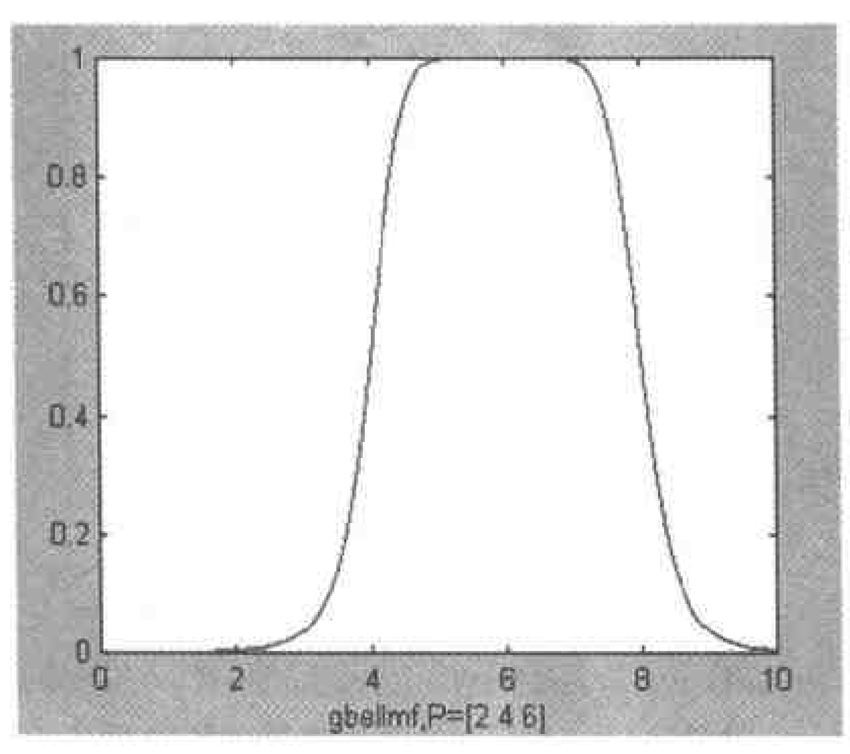
$$f(x;a,b,c)=\frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$$

将向量 params(元素分别为 a,b,c)作为 gbellmf 的第二个参数输入函数 gbellmf 中。其中参数 b 通常为正数,参数 c 将位于曲线正中。

例如：

```
x=0:0.1:10;
y=gbellmf(x,[2 4 6]);
plot(x,y)
xlabel('gbellmf,P=[2 4 6]')
```

得到的隶属函数如附图 B-4 所示。



附图 B-4　gbellmf,钟形隶属函数

参见：

dsigmf,gaussmf,gauss2mf,evalmf,mf2mf,pimf,psigmf,sigmf,smf,trapmf,trimf,zmf

5. pimf

(1) 功能

模糊逻辑工具箱内置的 π 形隶属函数。

(2) 使用格式

```
y=pimf(x,[a b c d])
```

(3) 说明

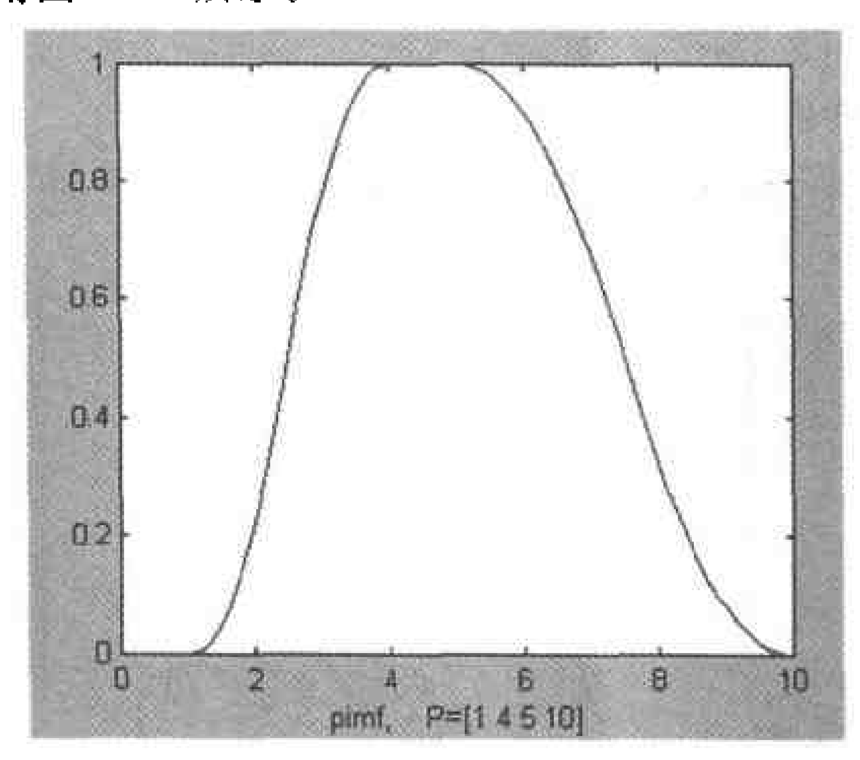
该隶属函数是基于样条的，其名称源于其形状像一个 π 字符。隶属函数在向量 x 的点上进行计算。参数 a、d 位于其隶属函数图形两足，参数 b、c 位于其隶属函数图形的两肩。

例如：

```
x=0:0.1:10;
y=pimf(x,[1 4 5 10]);
```

```
plot(x,y)
xlabel('pimf, P=[1 4 5 10]')
```

得到的隶属函数如附图 B-5 所示。



附图 B-5　pimf,形隶属函数

参见：

dsigmf, gaussmf, gauss2mf, gbellmf, evalmf, mf2mf, psigmf, sigmf, smf, trapmf, trimf, zmf

6. psigmf

(1) 功能

模糊逻辑工具箱内置隶属函数,它是由两个 sigmoid 函数的乘积构成的隶属函数。

(2) 使用格式

```
y=psigmf(x,[a1 c1 a2 c2])
```

(3) 说明

sigmoid 隶属函数在向量 x 的点上进行计算,其曲线图形由两个参数 a,c 决定,并由下式给出

$$f(x;a,c)=\frac{1}{1+e^{-a(x-c)}}$$

psigmf 是两个在向量 x 上计算的 sigmoid 隶属函数的简单相乘,即

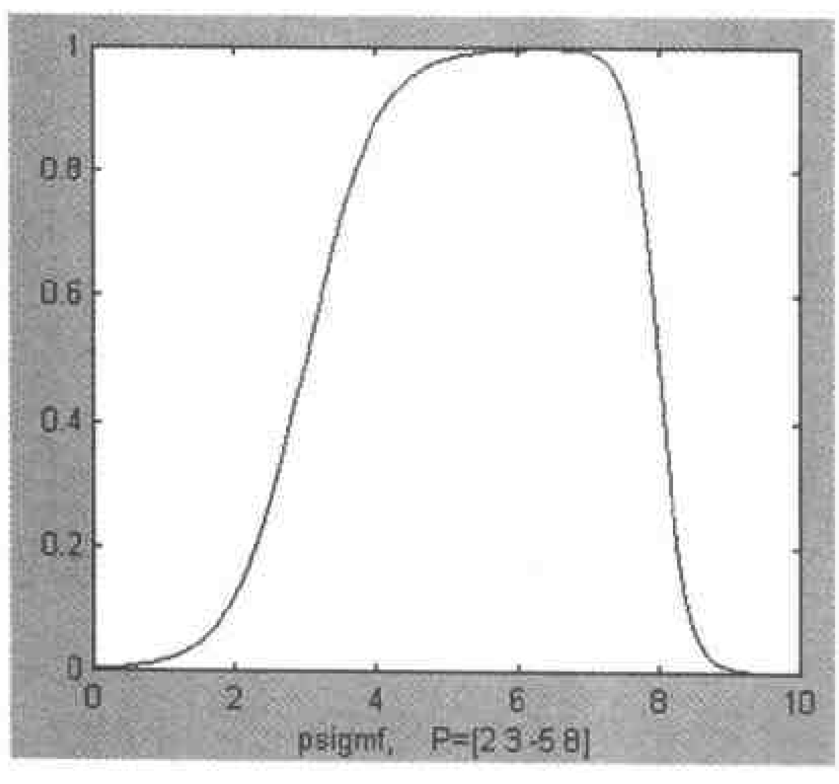
$$f_1(x;a_1,c_1)*f_2(x;a_2,c_2)$$

在函数 psigmf 中,参数的顺序为:[a1 c1 a2 c2]。

例如：

```
x=0:0.1:10;
y=psigmf(x,[2  3  -5  8]);
plot(x,y)
xlabel('psigmf, P=[2 3 -5 8]')
```

得到的隶属函数如附图 B-6 所示。



附图 B-6　psigmf,两个 sigmoid 函数的乘积构成的隶属函数

参见:

dsigmf,gaussmf,gauss2mf,gbellmf,evalmf,mf2mf,pimf,sigmf,smf,trapmf,trimf,zmf

7. smf

(1) 功能

模糊逻辑工具箱内置的 S 形隶属函数。

(2) 使用格式

```
y=smf(x,[a b])
```

(3) 说明

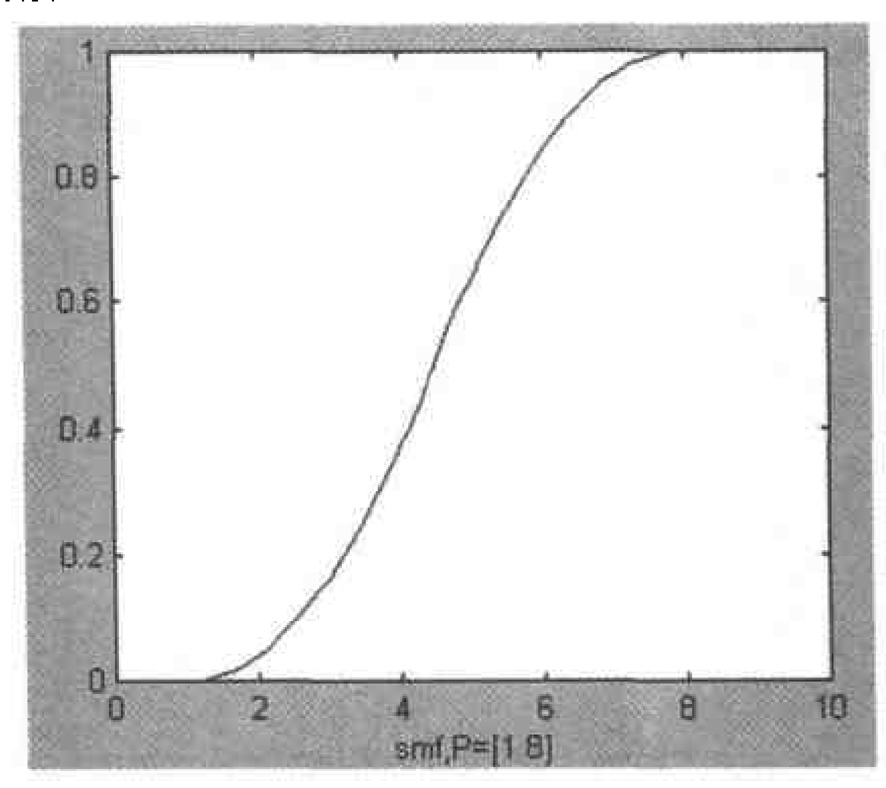
smf 隶属函数是对向量 x 映射的样条基曲线,其名称来源于其形状像一个 S 字符。参数 a,b 分别位于曲线倾斜部分的顶端。

例如:

```
x=0:0.1:10;
y=smf(x,[1 8]);
plot(x,y)
```

```
xlabel('smf,P=[1 8]')
```

得到的隶属函数如附图 B-7 所示。



附图 B-7　smf,S形隶属函数

参见：

dsigmf, gaussmf, gauss2mf, gbellmf, evalmf, mf2mf, pimf, psigmf, sigmf, trapmf, trimf, zmf

8. sigmf

(1) 功能

模糊逻辑工具箱内置的 Sigmoid 型隶属函数。

(2) 使用格式

```
y=sigmf(x,[a c])
```

(3) 说明

sigmf 函数是对矢量 x 的一个映射，它由两个参数 a 和 c 决定，并由下式给出

$$f(x;a,c)=\frac{1}{1+e^{-a(x-c)}}$$

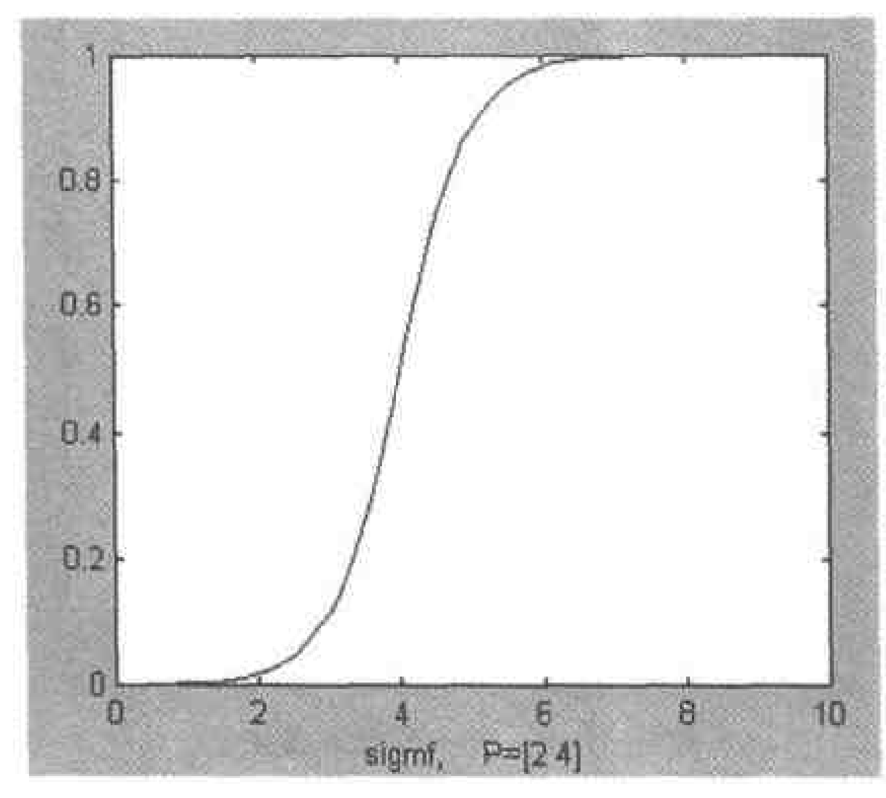
sigmoidal 隶属函数向右开口或向左开口由 a 的正负决定，并由此表示“极大”或“极小”的概念。更常见的隶属函数由两个 sigmoidal 隶属函数的积或差构成，参见 dsigmf 和 psigmf 隶属函数。

例如：

```
x=0:0.1:10;
y=sigmf(x,[2 4]);
```

```
plot(x,y)
xlabel('sigmf, P=[2 4]')
```

得到的隶属函数如附图 B－8 所示。



附图 B－8 sigmf,Sigmoid 隶属函数

参见：

dsigmf,gaussmf,gauss2mf,gbellmf,evalmf,mf2mf,pimf,psigmf,smf,trapmf,trimf,zmf

9. trapmf

(1) 功能

模糊逻辑工具箱内置的梯形隶属函数。

(2) 使用格式

```
y=trapmf(x,[a b c d])
```

(3) 说明

梯形曲线是关于向量 x 的函数，它由四个参数 a,b,c 和 d 参数决定，并由下式给出

$$f(x;a,b,c,d)=\begin{cases} 0, & x<a \\ \dfrac{x-a}{b-a}, & a\leqslant x\leqslant b \\ 1, & b\leqslant x\leqslant c \\ \dfrac{d-x}{d-c}, & c\leqslant x\leqslant d \\ 0, & d<x \end{cases}$$

或下面更紧凑的形式给出

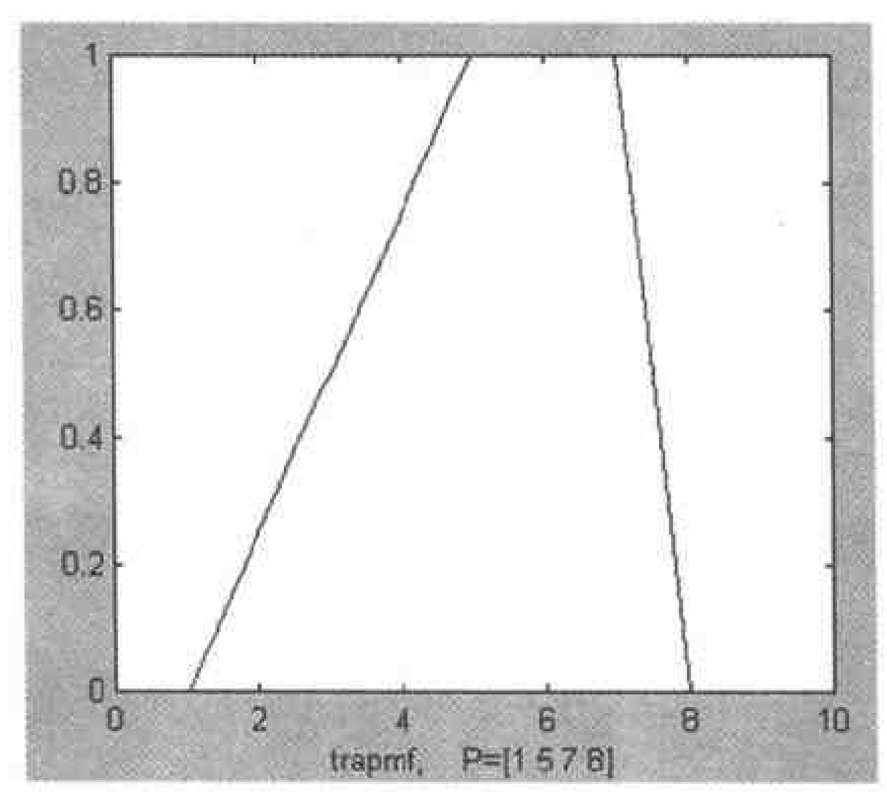
$$f(x;a,b,c,d)=\max\left\{\min\left\{\frac{x-a}{b-a},1,\frac{d-x}{d-c}\right\},0\right\}$$

其中参数 a,d 位于隶属函数图形的两足，参数 b,c 位于隶属函数图形的两肩。

例如：

```
x=0:0.1:10;
y=trapmf(x,[1 5 7 8]);
plot(x,y)
xlabel('trapmf, P=[1 5 7 8]')
```

得到的隶属函数如附图 B-9 所示。



附图 B-9 trapmf，梯形隶属函数

参见：

dsigmf，gaussmf，gauss2mf，gbellmf，evalmf，mf2mf，pimf，psigmf，sigmf，smf，trimf，zmf

10. trimf

(1) 功能

模糊逻辑工具箱内置的三角形隶属函数。

(2) 使用格式

```
y=trimf(x,params)
y=trimf(x,[a b c])
```

(3) 说明

trimf 隶属函数是关于向量 x 的函数，它由三个参数 a，b 和 c 决定，并由下式给出：

$$f(x;a,b,c)=\begin{cases}0, & x<a\\ \dfrac{x-a}{b-a}, & a\leqslant x\leqslant b\\ 1, & b\leqslant x\leqslant c\\ \dfrac{c-x}{c-b}, & b\leqslant x\leqslant c\\ 0, & c<x\end{cases}$$

或下面更紧凑的形式给出

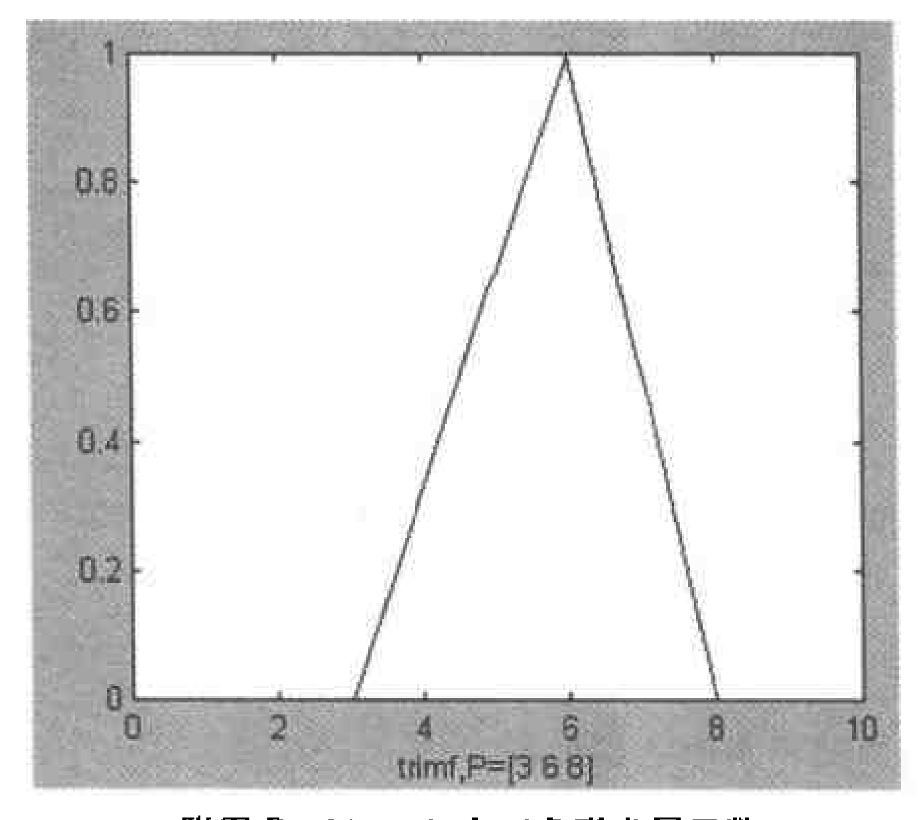
$$f(x;a,b,c)=\max\left\{\min\left\{\frac{x-a}{b-a},1,\frac{c-x}{c-b}\right\},0\right\}$$

其中参数 a,c 位于隶属函数曲线的两足,b 位于隶属函数曲线的顶端。

例如:

```
x=0:0.1:10;
y=trimf(x,[3 6 8]);
plot(x,y)
xlabel('trimf,P=[3 6 8]')
```

得到的隶属函数如附图 B-10 所示。



附图 B-10　trimf,三角形隶属函数

参见:

dsigmf,gaussmf,gauss2mf,gbellmf,evalmf,mf2mf,pimf,psigmf,sigmf,smf,trapmf

11. zmf

(1) 功能

模糊逻辑工具箱内置的 Z 形隶属函数。

(2) 使用格式

y=zmf(x,[a b])

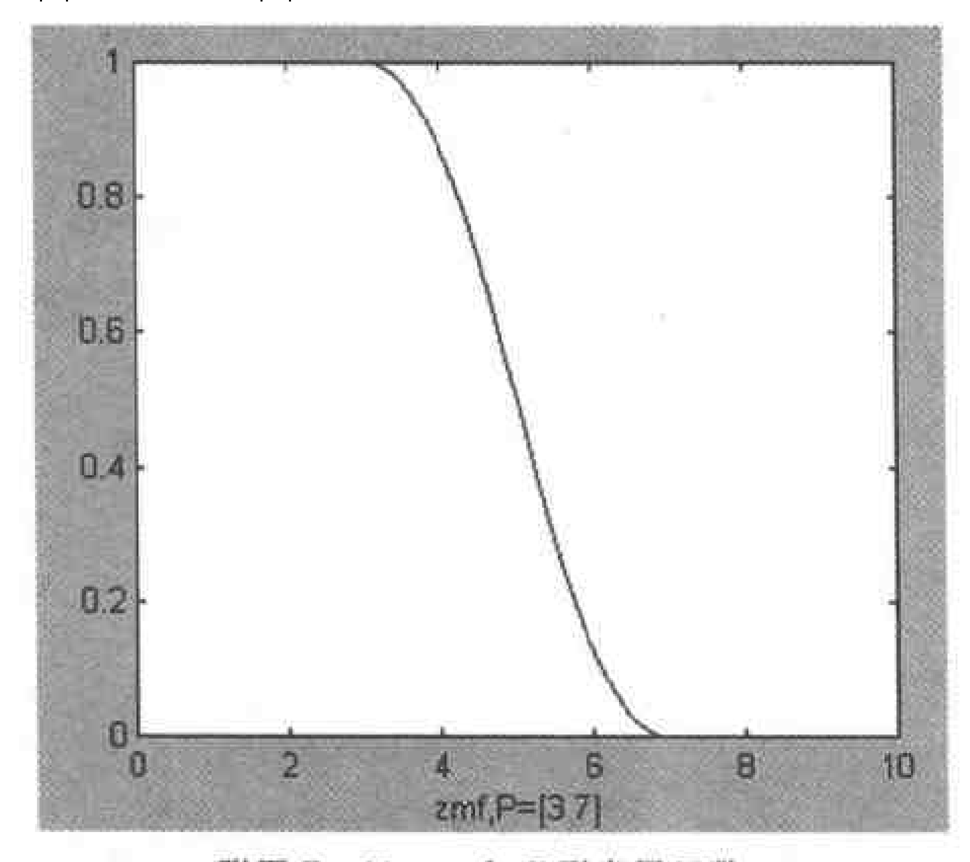
(3) 说明

zmf 隶属函数是关于向量 x 的函数,其名称来源于其形状像一个 Z 字符。其中参数 a,b 位于隶属函数曲线倾斜部分的顶端。

例如:

```
x=0:0.1:10;
y=zmf(x,[3 7]);
plot(x,y)
xlabel('zmf,P=[3 7]')
```

得到的隶属函数如附图 B-11 所示。



附图 B-11　zmf,Z 形隶属函数

参见:

dsigmf, gaussmf, gauss2mf, gbellmf, evalmf, mf2mf, pimf, psigmf, sigmf, smf, trapmf, trimf

C. FIS 数据结构管理

函数	功能
addmf	对模糊推理系统添加隶属函数
addrule	对模糊推理系统添加规则
addvar	对模糊推理系统添加变量
defuzz	对隶属函数进行反模糊化
evalfis	执行模糊推理计算
evalmf	一般的隶属函数估计
gensurf	产生模糊推理系统输出曲面
getfis	获得模糊系统属性
mf2mf	在函数之间转换参数
newfis	新建模糊推理系统
parsrule	解析模糊规则
plotfis	显示模糊推理系统输入变量结构
plotmf	绘制所有与某一给定变量相关的隶属函数
readfis	从磁盘载入模糊推理系统
rmmf	从模糊推理系统删除隶属函数
rmvar	从模糊推理系统删除变量
setfis	设置模糊系统属性
showfis	显示注解的模糊推理系统
showrule	显示模糊推理系统的规则
writefis	将模糊推理系统保存到磁盘

1. addmf

(1) 功能

对模糊推理系统添加隶属函数。

(2) 使用格式

a=addmf(a,'varType',varIndex,'mfName','mfType',mfParams)

(3) 说明

addmf 只能对在 MATLAB 工作区已存在的模糊推理系统添加隶属函数。

在对变量添加隶属函数时对各隶属函数指定了指针，因此对变量添加的第一个隶属函数称为该变量的第一隶属函数。

如果一个系统只定义了一个输入，就不能对尚不存在的第二个输入添加隶属函数。

addmf 函数需要按如下顺序的六个输入参数：

a，变量名，它是在 MATLAB 工作区中已存在的模糊推理系统结构变量。

'varType'，字符串'out'或'input'，它表示用户想要的对其添加隶属函数的变量类型。

varIndex，变量指针，它表示用户想要对其添加隶属函数的输入变量或输出变量。

'mfName'，字符串，它表示新建隶属函数的名称。

'mfType'，字符串，它表示新建隶属函数的类型。

mfParams，参数矢量，它指定隶属函数的参数。

参见：

addrule，addvar，plotmf，rmmf，rmvar

2. addrule

(1) 功能

对一个模糊推理系统添加模糊规则。

(2) 使用格式

a=addrule(a，ruleList)

(3) 说明

addrule 有两个参数。第一个参数是 MATLAB 工作区中的模糊推理系统变量名。第二个参数是一个单行或多行矩阵，每行表示一个给定的规则。

规则列表矩阵的格式是明确的。如果在系统中有 m 个输入和 n 个输出，则模糊规则列表的列数必须为 $m+n+2$。前面 m 列指向系统的 m 个输入，每列中包含一个数字，它指向该变量的隶属函数指针；后面 n 列指向系统的 n 个输出，每列中包含一个数字，同样，它也指向该变量的隶属函数指针。

第 $m+n+1$ 列包含应用相应规则的权值。权值必须在 0 和 1 之间，通常为 1。

第 $m+n+2$ 列表示对相应的规则前提的模糊运算。如果为 1，它表示为 AND 运算，如果为 2，它表示为 OR 运算。

参见：

addmf，addvar，rmmf，rmvar，parsrule，showrule

3. addvar

(1) 功能

对模糊推理系统添加变量。

(2) 使用格式

a=addvar(a,'varType','varName',varBounds)

(3) 说明

addvar 函数需要按如下顺序的四个输入参数：

a，变量名，表示在 MATLAB 工作区中的模糊推理系统结构。

'varType'，字符串'out'或'input'，表示想要添加的输入变量或输出变量的类型。

'varName'，字符串，表示用户想要添加的变量名称。

varBounds，向量，它描述了用户想要添加的变量的取值范围。

在对系统添加变量时对各变量指定了指针，因此对系统添加的第一个变量称为该系统的第一变量。

输入变量和输出变量的编号是相互独立的。

参见：

addmf,addrule,rmmf,rmvar

4. defuzz

(1) 功能

对隶属函数进行反模糊化。

(2) 使用格式

out=defuzz(x,mf,type)

(3) 说明

defuzz(x,mf,type)根据参数及其类型，对于变量 x 使用某一反模糊化方法，按隶属函数 mf 返回一个反模糊输出值。

参数 type 为下列类型之一：

centroid，区域矩心方法。

bisector，区域二等分方法。

mom，平均最大值方法。

som，最小最大值方法。

lom，最大最大值方法。

如果参数 type 不是上面五种之一，则系统假定它是一个自定义函数。x 和 mf 被传递给该函数，并产生一个反模糊化输出。

5. evalfis

(1) 功　能

执行模糊推理计算。

(2) 使用格式

```
output=evalfis(input,fismat)
output=evalfis(input,fismat,numPts)
[output,IRR,ORR,ARR]=evalfis(input,fismat)
[output,IRR,ORR,ARR]=evalfis(input,fismat,numPts)
```

(3) 说明

①evalfis 函数需要下列参数：

input,指定输入值的数字或矩阵。如果输入是一个 $M\times N$ 矩阵,其中 N 是输入变量的个数,则 evalfis 将输入每行作为一个输入矢量,并返回一个 $M\times L$ 矩阵到输出变量 output。

fismat,要进行计算的模糊推理系统。

numPts,可选参数项,它表示隶属函数在输入或输出范围上的采样点数。如果不采用该项,则 evalfis 采用其缺省值为 101 点。

②evalfis 的输出如下：

output,$M\times L$ 输出矩阵,其中,M 表示上面指定的输入变量的个数,L 表示模糊推理系统输出变量的个数。

只有在输入参数是一个行矢量时,才能对可选输出变量进行计算。这些可选输出变量如下：

IRR,通过隶属函数计算的输入值结果。它是一个 numRules$\times N$ 矩阵。其中 numRules 为规则个数,N 为输入变量的个数。

ORR,通过隶属函数计算的输出值结果。它是一个 numPts$\times$numRules$\times L$ 矩阵,其中 numRules 是规则的个数,L 是输出的个数。这个矩阵的前 numRules 行对应于第一个输出,下 numRules 行对应于第二个输出,等等。

ARR,这是一个 numPts$\times L$ 矩阵,它是在 numPts 上对每个输出的输出范围上的聚类采样。

当仅用一个输出变量调用 evalfis 函数时,这个函数计算由结构 fismat 指定的模糊推理系统在由输入数字或输入矩阵 input 指定输入时的输出矢量 output。

参见：

ruleview,gensurf

6. evalmf

(1) 功能

通用隶属函数计算。

(2) 使用格式

y=evalmf(x,mfParams,mfType)

(3) 说明

evalmf 可对任何隶属函数进行计算,其中 x 是对隶属函数计算的变量范围,mfType 是工具箱的某个隶属函数,mfParams 是该隶属函数的适当参数。

evalmf 同样可以进行自定义隶属函数的工作,因为对任何它无法识别名称的隶属函数,它同样可以进行计算。

参见:

dsigmf,gaussmf,gauss2mf,gbellmf,mf2mf,pimf,psigmf,sigmf,smf,trapmf,trimf,zmf

7. gensurf

(1) 功能

产生模糊推理系统的输出曲面。

(2) 使用格式

gensurf(fis)

gensurf(fis,inputs,output)

gensurf(fis,inputs,output,grids,refinput)

(3) 说明

对一个给定的使用前两个输入和第一个输出的 FIS 系统,用 gensurf(fis)绘制其模糊推理的输出曲面。

当输入由向量 input 给定(一个或两个),输出由标量 output 给定时,用 gensurf(fis,inputs,output)绘制出模糊推理系统的输出曲面。

当在 X(水平)方向和 Y(垂直)方向上指定网格数时,用 gensurf(fis,inputs,output,grids)绘制模糊推理系统的输出曲面。如果网格是二维矢量,则在 X 和 Y 方向的网格可以独立设置。

如果输入多于两个,可以使用 gensurf(fis,inputs,output,grids,refinput),其中 refinput 指定系统的不变输入。

[x,y,z]=gensurf(...)返回定义输出曲面和抑制自动绘图的变量。

参见:

evalfis,surfview

8. getfis

(1) 功能

获取模糊系统属性。

(2) 使用格式

getfis(a)

getfis(a,'fisprop')

getfis(a,'vartype',varindex,'varprop')

getfis(a,'vartype',varindex,'mf',mfindex)

getfis(a,'vartype',varindex,'mf',mfindex,'mfprop')

(3) 说明

getfis 是访问模糊推理系统的基函数。使用该函数可以了解模糊推理系统各部分的信息。

getfis 各参数的意义如下：

a,工作区中模糊推理系统的变量名。

'vartype'，字符串，表示想要获取的输入变量或输出变量的类型。

varindex，整数，表示要显示想要获得的变量的指针(例如,1 为 input1)。

'mf'，必需的字符串，表示要搜索的是隶属函数的信息。

mfindex，要查找其信息的隶属函数的指针。

参见：

setfis,showfis

9. mf2mf

(1) 功能

转换两个隶属函数之间的参数。

(2) 使用格式

outParams=mf2mf(inParams,inType,outType)

(3) 说明

mf2mf 函数根据隶属函数的参数设置，可以将任一内置隶属函数转换为另一个隶属函数。原则上，mf2mf 在新隶属函数和原隶属函数之间对参数进行点对点的转换。但有时在转换中会丢失信息，因此如果将输出的参数再转换到原来的隶属函数类型中，二次转换后的隶属函数与其原隶属函数形状会不一样。

mf2mf 的输入参数如下：

inParams，被转换隶属函数的参数。

inType，字符串，表示被转换隶属函数的类型。

outType，字符串，表示新隶属函数的类型。

参见：

dsigmf，gaussmf，gauss2mf，gbellmf，evalmf，pimf，psigmf，sigmf，smf，trapmf，trimf，zmf

10. newfis

(1) 功能

新建一个模糊推理系统。

(2) 使用格式

a＝newfis(fisName，fisType，andMethod，orMethod，impMethod，... aggMethod，defuzzMethod)

(3) 说明

该函数建立一个新的模糊推理系统结构。newfis 函数使用模糊推理系统中的七个输入参数和一个输出参数。

该函数的七个输入参数如下：

fisName，字符串，将要建立的模糊推理系统结构名称。

fisType，将要建立模糊推理系统的类型。

andMethod，orMethod，impMethod，aggMethod 和 defuzzMethod 分别提供 AND，OR，蕴涵，聚类和反模糊化的方法。

参见：

readfis，writefis

11. parsrule

(1) 功能

解析模糊规则。

(2) 使用格式

fis2＝parsrule(fis，txtRuleList)

fis2＝parsrule(fis，txtRuleList，ruleFormat)

fis2＝parsrule(fis，txtRuleList，ruleFormat，lang)

(3) 说明

parsrule 函数解析 MATLAB 工作区中以模糊推理系统变量.fis 进行规则定义的文本(txtRuleList)，并返回一个带有适当规则列表的模糊推理系统结构。

如果初始输入的模糊推理系统结构.fis 带有任何的初始规则，在新建结构.fis2 中它们将被取代。

parsrule 函数支持三种不同格式的规则(由 ruleFormat 表示):verbose,symbolic 和 indexed, 默认格式为 verbose。当使用了语言参数可选项 lang 时,规则以 verbose 格式被解析,并采用语言 lang 中的关键词。语言 lang 必须是 English,Francais,Deutsch 中的一种。当使用英语时,关键词是 if,then,is,AND,OR 和 NOT。

参见:

addrule,ruleedit,showrule

12. plotfis

(1) 功能

绘制模糊推理系统的图形。

(2) 使用格式

plotfis(fismat)

(3) 说明

plotfis 函数绘制出模糊推理系统 fismat 的高级图形。输入及其隶属函数显示在模糊推理系统结构特征的左边,同时输出及其隶属函数显示在模糊推理系统结构特征的右边。

参见:

plotfis,evalmf

13. plotmf

(1) 功能

绘制给定变量所有的隶属函数。

(2) 使用格式

plotmf(fismat,varType,varIndex)

(3) 说明

plotmf 函数绘制在模糊推理系统结构 fismat 中与某个给定变量相关的所有的隶属函数。变量分别由 varType(input 或 output)和 varIndex 给定。

plotmf 函数同样可用于 MATLAB 函数 subplot 中。

参见:

evalmf,plotfis

14. readfis

(1) 功能

从磁盘中载入一个模糊推理系统。

(2) 使用格式

fismat=readfis('filename')

(3) 说明

readfis 函数从磁盘上的.fis 文件读取模糊推理系统，并将文件结果返回到工作区中。

fismat=readfis(无输入参数)将打开一个统一格式的打开文件对话框，以对文件路径和名称定位。

参见：

writefis

15. rmmf

(1) 功能

从模糊推理系统中删除隶属函数。

(2) 使用格式

fis=rmmf(fis,'varType',varIndex,'mf',mfIndex)

(3) 说明

rmmf 函数从模糊推理系统.fis 中删除变量类型为 varType(input 或 output)、指针为 varIndex 的隶属函数 mfIndex。其中参数 mf 是必需的，它是代表隶属函数的字符串。

参见：

addmf,addrule,addvar,plotmf,rmvar

16. rmvar

(1) 功能

从模糊推理系统中删除变量。

(2) 使用格式

fis2=rmvar(fis,'varType',varIndex)

(3) 说明

[fis2,errorStr]=rmvar(fis,'varType',varIndex)从模糊推理系统相关工作区的 FIS 结构.fis中删除类型为 varType(input 或 output)、指针为 varIndex 的变量。

[fis2,errorStr]=rmvar(fis,'varType',varIndex)将所有的错误信息返回到字符串 errorStr 中。

rmvar 函数自动地改变模糊规则列表，使其与当前变量的个数保持一致。在删除一个变量之前，必须删除模糊推理系统的包含了准备删除变量的所有规则。对当前模糊规则列表中正在使用的模糊变量，不能予以删除。

参见：

addmf,addrule,addvar,rmmf

17. setfis

(1) 功能

设置模糊系统属性。

(2) 使用格式

a=setfis(a,'fispropname','newfisprop')

a=setfis(a,'vartype',varindex,'varpropname','newvarprop')

a=setfis(a,'vartype',varindex,'mf',mfindex,'mfpropname','newmfprop');

(3) 说明

可以用三个、五个或七个输入参数调用 setfis 命令。这取决于要对一个属于模糊推理系统的特定变量、还是对一个属于某一变量的特定隶属函数设置整个模糊推理系统结构的参数。

setfis 的参数如下：

a，工作区中一个模糊推理系统变量名。

'vartype'，字符串，表示变量类型('input'或'output')。

varindex，输入或输出变量指针。

'mf'，字符串，在用七个参数调用 setfis 时，它是必需的第四个参数，表示该变量是一个隶属函数。

mfindex，属于一个选定变量的隶属函数指针。

'fispropname'，字符串，表示想要设置的模糊推理系统的属性，如 name，type，andmethod，ormethod，impmethod，aggmethod，defuzzmethod。

'newfisprop'，字符串，它描述模糊推理系统的属性，或想要使用的方法。

'varpropname'，字符串，它指明想要设置的变量域名称 name 或 range。

'newvarprop'，字符串，表示想要的变量名称(对 name)，或一个变量范围描述的数组(对 range)。

'mfpropname'，字符串，表示想要设置的隶属函数域的 name，type 或 params。

'newmfprop'，字符串，表示用户想要设置的隶属函数域的名称或类型(对 name 或 type)，或一个描述参数范围的数组(对 params)。

参见：

getfis

18. showfis

(1) 功能

显示注释性模糊推理系统。

(2) 使用格式

showfis(fismat)

(3) 说明

showfis 命令打印出带注释的模糊推理系统变量 fismat(该系统变量应当已在工作区中)，从而可以看到系统结构中每一部分的内容和意义。

参见：

getfis

19. showrule

(1) 功能

显示模糊推理系统的模糊规则。

(2) 使用格式

showrule(fis)

showrule(fis,indexList)

showrule(fis,indexList,format)

showrule(fis,indexList,format,lang)

(3) 说明

使用 showrule 命令可以显示与给定系统相关的模糊规则。

可以用一到四个参数调用 showrule 命令。

第一个参数 fis 是必需的。它是 MATLAB 工作区中一个模糊推理系统结构的变量名。

第二个参数 indexList(可选)是用户想要显示的模糊规则列表。

第三个参数 format(可选)是一个字符串，表示返回规则的格式。Showrule 以三种不同格式中的任何一种返回规则。这三种格式为 'verbose'(默认方式)、'symbolic' 和 'indexed'。

如果要使用第四个参数 lang 调用 showrule，则第三个参数必须是 verbose，且 showrule(fis,indexList,format,lang)按参数 lang 指定的语言(必须为 english，francais 或 deutsch 之一)显示模糊规则。

参见：

parsrule，ruleedit，addrule

20. writefis

(1) 功能

将模糊推理系统保存到磁盘。

(2) 使用格式

writefis(fismat)

writefis(fismat,'filename')

writefis(fismat,'filename','dialog')

(3) 说明

writefis 将 MATLAB 工作区的模糊推理系统结构 fismat 保存到磁盘上。保存文件的扩展名为.fis。

writefis(fismat)打开一个对话框,以对保存文件的路径和名称定位。

writefis(fismat,'filename')将相应于模糊推理系统结构 fismat 的.fis 文件按 filename.fis 的名称写到磁盘上。这种情况下不打开对话框,而将文件保存到当前目录。

writefis(fismat,'filename','dialog')应用默认的文件名 filename.fis,并打开一个对话框。

如果在文件名中没有包括扩展名.fis,writefis 命令自动地将扩展名加到文件之后。

参见:

readfis

D. 高级技术

函数	功能
anfis	Sugeno 型模糊推理系统的训练程序(仅用于 MEX 类型)
fcm	使用模糊 C 均值聚类法找到聚类
genfis1	使用网格法产生模糊推理系统矩阵
genfis2	使用消去聚类法产生模糊推理系统矩阵
subclust	使用消去聚类法找到聚类中心

1. anfis

(1) 功能

Sugeno 型模糊推理系统的训练程序(仅用于 MEX 类型)

(2) 使用格式

```
[fismat,error1,stepsize]=anfis(trnData)
[fismat,error1,stepsize]=anfis(trnData,fismat)
[fismat1,error1,stepsize]=...
anfis(trnData,fismat,trnOpt,dispOpt)
[fismat1,error1,stepsize,fismat2,error2]=...
anfis(trnData,trnOpt,dispOpt,chkData)
[fismat1,error1,stepsize,fismat2,error2]=...
anfis(trnData,trnOpt,dispOpt,chkData,optMethod)
```

(3) 说明

anfis 是 Sugeno 型模糊推理系统主要的训练程序。

anfis 是用混合学习算法辨识 Sugeno 型模糊推理系统的参数。在用训练模糊推理系统的隶属函数参数模仿一个给定的训练数据集时,anfis 应用的是最小二乘法和反向传播梯度下降算法相结合的方法。

在模型确认中还可以用一个可选参数调用 anfis,伴随该选项的模型确认类型是对模型完全匹配的检查。该参数是一个数据集合,称为检验数据集。

在使用格式中涉及的 anfis 参数:

①trnData:训练数据集名称,这是一个矩阵,除最后一列为包含了一个输出数据的单矢量

外，在其余各列中包含了所有的输入数据。

②fismat：模糊推理系统名称。在训练过程中，该模糊推理系统向 anfis 提供一个初始隶属函数集。如果不使用该选项，则在训练过程中 anfis 使用 genfis1 产生一个默认的初始模糊推理系统。如果只用一个参数调用 anfis，这个默认的模糊推理系统中将带有两个高斯型隶属函数。

如果参数 fismat 的形式是一个单独的数字，则 anfis 将其视为隶属函数的个数来使用；如果参数 fismat 的形式是一个矢量，则 anfis 将其每一元素分别作为与不同输入相关联的隶属函数的个数来使用。这种情况下，在开始训练过程前，anfis 的两个参数都被传递给 genfis1，以产生一个有效的模糊推理系统结构。

③trnOpt：训练选项向量。当该向量中的任何项输入值为 NaN(not a number)时，将强制使用选项的默认值。这些选项如下：

trnOpt(1)，训练时段数(默认值：10)。

trnOpt(2)，训练误差目标(默认值：0)。

trnOpt(3)，初始步长(默认值：0.01)。

trnOpt(4)，步长减小速度(默认值：0.9)。

trnOpt(5)，步长增加速度(默认值：1.1)。

④dispOpt：显示选项向量。它指定了在训练过程中要在 MATLAB 命令窗口显示的信息。所有显示选项的默认值均为 1，即表示所有相应信息都将被显示。如果将某一项设置为 0，则表示不在屏幕上显示相应选项的信息。对任何显示选项，当其输入为 NaN 时，将强制使用选项的默认值。这些选项如下：

dispOpt(1)，ANFIS 信息，例如输入隶属函数与输出隶属函数个数(默认值：1)。

dispOpt(2)，误差(默认值：1)。

dispOpt(3)，每一参数更新时的步长(默认值：1)。

dispOpt(4)，最终结果(默认值：1)。

⑤chkData：在完全匹配模型确认时，anfis 可选检验数据集的名称。该数据集是一个矩阵，其格式与训练数据集格式相同。

⑥optMethod：在隶属函数参数训练中使用的可选优化方法。当 optMethod 值为 1 时，使用混合法；当 optMethod 值为 0 时，使用反向传播算法。默认方法是混合法，它是最小二乘法估计与反向传播算法的结合。当参数 optMethod 被设置为任何非 0 的值时，anfis 都将按 optMethod 的默认值使用。

任何情况下，只要达到训练的时间长度，或达到训练误差目标，则 anfis 停止训练过程。

⑦anfis：当用两个或更多个参数调用 anfis 时，对任何可选参数项，如果其输入值为 NaN 或空矩阵，anfis 将使用该项的默认值。默认值可以在文件 anfis.m 中直接进行修改。除了要

指定的参数项(例如要实现 anfis 的检验数据选项),对不想指定默认值的参数项,可以用 NaN 或空矩阵占据变量位置。

在使用格式中,anfis 的各输出项意义如下:

fismat1,模糊推理系统结构,其参数根据最小训练误差标准设置。

error1 或 error2,分别表示训练误差数据信号和检验误差数据信号的均方根误差。

stepsize,步长数组。如果误差测量连续两次一增一减,则步长乘以一个小于 1 的常量,以减小步长;如果误差测量连续四次减小,则步长乘以一个大于 1 的常量,以增加步长。

fismat2,模糊推理系统结构,其参数根据最小检验误差标准设置。

参见:

genfis1.anfis

2. fcm

(1) 功能

模糊 C 均值聚类。

(2) 使用格式

[center,U,obj_fcn]=fcm(data,cluster_n)

(3) 说明

对一个给定的数据集 data,命令 fcm 应用模糊 C 均值聚类方法。

fcm 的输入参数如下:

data,要进行聚类的数据集。每行是一个采样数据点。

cluster_n,聚类个数(大于 1)。

fcm 的输出参数如下:

center,结果聚类中心矩阵,其中每一行提供一个中心坐标。

U,结果模糊分区矩阵(或隶属函数矩阵)。

obj_fcn,迭代过程中的目标函数值。

此外,还可以使用另一个可选参数项 options 调用 fcm,以控制聚类参数。其形式为"fcm(data,cluster_n,options)"。它引入一个停止训练标准并(或)设置迭代信息的显示。Options 的各选项如下:

options(1),分区矩阵 U 的指数(默认值:2.0)。

options(2),最大迭代次数(默认值:100)。

options(3),最小改进量(默认值:1e-5)。

options(4),显示迭代过程中的信息(默认值:1)。

对这些选项中的任一项,如果其输入为 NaN,则将使用该项的默认值。

当达到最大迭代次数，或连续两次迭代结果小于指定的最小改进量，fcm 就停止聚类过程。

3. genfis1

(1) 功能

不使用数据聚类而从数据中产生一个模糊推理系统结构。

(2) 使用格式

fismat=genfis1(data)

fismat=genfis1(data,numMFs,inmftype,outmftype)

(3) 说明

genfis1 产生一个 Sugeno 型模糊推理系统结构，以作为 anfis 训练的初始条件(即隶属函数参数的初始化)。

genfis1(data,numMFs,inmftype,outmftype)使用一个数据上的网格分区(无聚类)，从训练数据集 data 上产生一个模糊推理系统结构。

genfis1 的输入参数如下：

data，训练数据矩阵。矩阵中除最后一列表示单个输出外，其它元素必须表示输入数据。

numMFs，向量。向量中的元素表示与每个输入相关的隶属函数的个数。如果与每个输入相关的隶属函数个数相等，则 numMFs 参数项可以是一个单个的数字。

inmftype，字符串数组。数组中的每一元素(字符串)指定与每个输入相关的隶属函数的类型。同样，如果与每个输入相关的隶属函数类型相同，则 inmftype 参数项可以是一个单字符串。

outmftype，字符串。它指定与输出相关的隶属函数类型。由于 genfis1 产生的是 Sugeno 型系统，因此只能有一个输出，并且输出隶属函数类型只能为线性的或常量。

与输出相关的隶属函数个数同 genfis1 产生的模糊规则个数相等。NumMFs 的默认值(即默认的隶属函数个数)为 2；输入或输出隶属函数的默认类型为 gbellmf。任何情况下，如果调用 genfis1 时没有使用后三个参数，则 genfis1 将使用这些默认值。

参见：

anfis

4. genfis2

(1) 功能

使用消去聚类法从数据中产生一个模糊推理系统结构。

(2) 使用格式

fismat = genfis2(Xin,Xout,radii)

fismat = genfis2(Xin,Xout,radii,xBounds)

fismat = genfis2(Xin,Xout,radii,xBounds,options)

(3) 说明

当分别给出输入数据集和输出数据集时,genfis2 使用模糊消去聚类法产生一个模糊推理系统。当只有一个输出时,可以首先对数据采用消去聚类法,用 genfis2 为 anfis 的训练过程产生一个初始模糊推理系统。genfis2 采用提取对数据行为建模的一个模糊规则集来完成此工作。而模糊规则的提取方法是,首先使用 subclust 函数,以决定模糊规则和前提隶属函数的个数;然后用线性最小方差估计决定每个模糊规则的结果方程;该方程返回一个模糊推理系统结构;在这个结构中包含了一个模糊规则集,以覆盖特征空间。

genfis2 的输入参数如下:

Xin,矩阵。其中每行包含一个数据点的输入值。

Xout,矩阵。其中每行包含一个数据点的输出值。

radii,向量。它假定数据都位于一个单位超集中,然后指定聚类中心在每一数据维上的影响范围。例如,如果数据维数为 3(即 Xin 为 2 列,Xout 为 1 列),radii=[0.5 0.4 0.3]分别指定聚类中心在第一、第二和第三维(即 Xin 的第一列、Xin 的第二列和 Xou 列)的影响范围是数据空间宽度的 0.5,0.4 和 0.3。如果 radii 是一个标量,则在所有数据维上都使用该标量值,即每个聚类中心的影响范围是一个给定半径的球形邻域。

xBounds, $2\times N$ 可选矩阵。它指定怎样将 Xin 和 Xout 中的数据映射到一个单位超集中。其中 N 为数据的维数(行)。Xbounds 的第一行中包含了在每一维上缩放数据的最小范围,第二行中包含在每一维上缩放数据的最大范围。例如,xBounds=[-10 0 -1; 10 50 1]指定将第一维数据上的值从范围[-10 +10]缩放到[0 1]上,将第二维数据上的值从范围[0 50]上缩放到[0 1]上,将第三维数据上的值从范围[-1 1]上缩放到[0 1]上。如果 genfis2 的 xBounds 参数项被设为一个空矩阵,或未被设置,则 genfis2 将 xBounds 参数默认为每一维数据维上的最小值和最大值。

options,可选向量。它指定算法参数,而不使用其默认值。这些参数的意义参见 subclust 的函数详解。当这个参数未被指定时,genfis2 将使用其默认值。

参见:

subclust

5. subclust

(1) 功能

利用消去聚类法估计聚类中心。

(2) 使用格式

[C,S] = subclust(X,radii,xBounds,options)

(3) 说明

该函数使用消去聚类法估计一个数据集的聚类中心。消去聚类法假定每一个数据点都可以是聚类中心,并在该数据点邻近数据密度的基础上,计算每一个数据点被定义为聚类中心的可能性。

对消去聚类算法的过程描述如下:

①选择具有最大可能性的数据点为第一聚类中心。

②删除第一聚类中心邻近的所有数据点(由 radii 决定),以便决定下一个数据聚类及其中心位置。

③迭代此过程,直到所有的数据都在聚类中心的半径 radii 以内。

消去聚类法是对 R. Yager 提出的上山聚类法的扩展。Subclust 的各输入参数如下:

X,矩阵。矩阵 X 中包含要进行聚类的数据,X 的每一行是一个数据点。

radii,向量。Radii 向量中的每一元素都在 0 和 1 之间,它假定数据都在一个单位超集中,并指定聚类对每一数据维的影响半径。如果 radii 的值较小,则通常会得到一些大的聚类。Radii 较为适当的值通常在 0.2 到 0.5 之间。例如,如果数据维数为 2(X 为 2 列),radii=[0.5 0.25],则指定聚类对第一维数据的影响范围是数据空间宽度的 1/2,对第二维数据的影响范围是数据空间宽度的 1/4。如果 radii 是一个标量,则对每一维上的数据都使用该标量值,即每个聚类中心的影响范围是一个给定半径的球形邻域。

xBounds,2×N 的可选矩阵。它指定怎样将 X 的数据映射到一个单位超集中。其中 N 为数据维数。如果 X 已被规范化,则参数 xBounds 是可选的。xBounds 的第一行中包含了在每一维上缩放数据的最小范围,第二行中包含在每一维上缩放数据的最大范围。例如,xBounds=[-10 0 -1; 10 50 1]指定将第一维数据上的值从范围[-10 +10]缩放到[0 1]上,将第二维数据上的值从范围[0 50]上缩放到[0 1]上,将第三维数据上的值从范围[-1 1]上缩放到[0 1]上。如果 xBounds 参数项被设为一个空矩阵,或未被设置,则将 xBounds 参数默认为每一维数据维上的最小值和最大值。

options,可选向量。可以用这个可选参数指定聚类算法参数,而不使用其默认值。该向量的各项元素如下:

options(1)=quashFactor。这个因子用以与 radii 值相乘,以决定聚类中心的邻域,从而消去无关点被考虑为聚类部分的可能性(默认值:1.25)。

options(2)=acceptRatio。它设置一个可能性,将其作为第一个聚类中心可能性的阈值。大于这个可能性的数据点,才可以被接受为一个聚类中心(默认值:0.5)。

options(3)=rejectRatio。它设置一个可能性,将其作为第一个聚类中心可能性的阈值。小于这个可能性的数据点,将被拒绝接受为一个聚类中心(默认值:0.15)。

options(4)=verbose。如果该项非零，则聚类过程中信息将作为聚类的结果信息被打印出来。(默认值:0)

函数 subcluster 在矩阵 C 中返回聚类中心。C 中每一行包含一个聚类中心的位置。在返回的 S 向量中包含 σ 值(均方根)，它指定一个聚类中心在每一维数据上的影响范围。所有的聚类中心共享一个相同的值集合。

参见：

genfis2

E. 仿真模块

函数	功能
fuzblock	模糊逻辑控制模块和演示模块
sffis	模糊推理 S 函数

1. fuzblock

(1) 功能

Simulink 模糊逻辑控制器模块。

(2) 使用格式

fuzblock

(3) 说明

fuzblock 命令除了打开一些 Simulink 演示模块外，还打开一个 Simulink 系统，该系统中包含两个可以使用的 Simulink 模块。

①The Fuzzy Logic Controller，模糊逻辑控制器。

②The Fuzzy Logic Controller With Rule Viewer，含规则观察器的模糊逻辑控制器。参见"模糊规则观察器"一节。在 Simulink 的仿真过程中，这个模块强制打开规则观察器。在模糊逻辑控制器模块上双击，可以打开与这些模块相关的对话框。在打开的对话框中，包含了工作区中与用户想要在 Simulink 模型中包含的模糊系统相对应的模糊推理系统结构名称。

要在带模糊规则观察器的模糊逻辑控制器打开这个对话框，请做如下两步：

①在模块上双击，打开一个带模糊逻辑控制器的 Simulink 图形。

②在第二个打开的模糊逻辑控制器上双击。

如果模糊推理系统有多个输入，在将其返回模糊逻辑控制器或带规则观察器的模糊逻辑控制器之前，这些输入将被相乘到一起。同样，如果系统有多个输出，这些信号将从一条相乘线上的模块上输出。

参见：

sffis，ruleview

2. sffis

(1) 功能

Simulink 的模糊推理 S 函数

(2) 使用格式

output=sffis(t,x,u,flag,fismat)

(3) 说明

Simulink 使用一个 MEX 文件来完成通常由 evalfis 执行的功能。在 Simulink 环境中,它是最优化工作的。就是说,与其它函数相比,sffis 在 Simulink 仿真的初始化阶段就在内存中建立一个数据结构,并在仿真结束之前一直使用该结构。

sffis 的输入参数 t,x 以及 flag 即为标准 Simulink S 函数的参数(请参看 Simulink 的相关文献),参数 u 为 MATLAB 工作区中模糊推理系统结构 fismat 的输入。例如,如果 fismat 有两个输入,则 u 将是一个 2 元向量。

参见:

evalfis, fuzblock

F. 演示程序

函数	功能
defuzzdm	反模糊化方法
fcmdemo	模糊 C 均值聚类演示(二维)
fuzdemos	模糊逻辑工具箱图形界面演示
gasdemo	使用消去聚类法的燃料效率 ANFIS 演示
juggler	带规则观察器的球体游戏
invkine	机械手反转运动
irisfcm	模糊 C 均值聚类演示(四维)
noisedm	自适应消噪演示
slbb	球体和横梁平衡控制 (Simulink)
slcp	反向钟摆控制(Simulink)
sltank	水位控制 (Simulink)
sltankrule	带规则观察器的水位控制(Simulink)
sltbu	倒车演示(只用于 Simulink)

fuzdemos

(1) 功能

列出所有的模糊逻辑工具箱演示。

(2) 使用格式

fuzdemos

(3) 说明

fuzdemos 函数打开一个图形工具,从该图形工具中,可以选择并运行上表中列出的任何模糊逻辑工具箱演示程序。

主要参考文献

[1] Wang H, Liu G P, Harris C J, Brown M. Advanced Adaptive Control. [s. l.]: Pergamon Press, 1996

[2] Fuzzy Logic Toolbox For Use with Matlab. The MathWorks, Inc. 1999

[3] 陈贻源. 模糊数学. 武汉:华中工学院出版社,1984

[4] 张文修,梁广锡. 模糊控制与系统. 西安:西安交通大学出版社,1998

[5] 李人厚. 智能控制理论和方法. 西安:西安电子科技大学出版社,1999

[6] 汪培庄,李洪兴. 模糊系统理论与模糊计算机. 北京:科学出版社,1996

[7] 陶永华等. 新型 PID 控制及其应用. 北京:机械工业出版社,1998

[8] 王顺晃等. 智能控制系统及其应用. 北京:机械工业出版社,1995

[9] 薛定宇. 控制系统计算机辅助设计——MATLAB 语言及应用. 北京:清华大学出版社,1997

[10] 刘漫丹,杜维. 单隶属度函数模糊控制器的研究. 仪器仪表学报,1999,(1):13~15

[11] 肖卫国,胡赤兵等. 基于模糊控制器的无静差跟踪控制方法研究. 机械工程学报,2000,36(1):24~26

[12] 张琰,纪运广. 模糊控制器在伺服系统中的应用. 电子技术应用,1999,(1):38~39

[13] 亢海伟,杨庆芬,王硕禾. 基于 MATLAB 模糊逻辑工具箱的模糊控制系统仿真. 电子技术应用,2000,(2):43~44

[14] 赵慧,韩俊伟,曾祥荣. 一种新型的模糊控制器设计方法. 黑龙江自动化技术与应用,1999,18(5):27~29

[15] 曲建岭,王磊,高峰. 水温模糊控制系统的设计. 自动化仪表,2001,21(2):8~13

[16] 刘革辉,单杰峰,郑楚光. MATLAB 软件中的模糊逻辑工具箱在模糊控制系统仿真中的应用. 计算机仿真,2000,(5):69~72

［Ｇｅｎｅｒａｌ Ｉｎｆｏｒｍａｔｉｏｎ］
书名＝模糊控制及其ＭＡＴＬＡＢ应用（张国良２００２．１１）
作者＝
页数＝１０００
ＳＳ号＝０
出版日期＝